DIGITAL SIMULATION AS AN APPROACH TO ESM SOFTWARE DEVELOPMENT

Robert Daniel Stanga

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DIGITAL SIMULATION AS AN APPROACH
TO ESM SOFTWARE DEVELOPMENT

by

Robert Daniel Stanga, Jr.

March 1975

Thesis Advisor: S. Jauregui

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Digital Simulation as an Approach to ESM Software Development

5. TYPE OF REPORT & PERIOD COVERED
Master's Thesis; March 1975

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Robert Daniel Stanga, Jr.

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE
March 1975

13. NUMBER OF PAGES
96

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Selective problem areas relating to computer based ESM system software development are discussed. Digital simulation is presented as a practical approach to solving software development problems. An algorithm to simulate a hard-wired EW signal preprocessor is developed. An application oriented high level language version of the algorithm is presented.
A FORTRAN program which generates realistic data for signal

analysis is described. Using the simulated data, an instruction level simulation program, and a language compiler, the algorithm is tested using an IBM 360 computer.

Digital Simulation as an Approach
to ESM Software Development

by

Robert Daniel Stanga, Jr.
Lieutenant, United States Navy
B.S.A.E., Auburn University, 1968

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1975

ABSTRACT

Selective problem areas relating to computer based ESM system software development are discussed. Digital simulation is presented as a practical approach to solving software development problems. An algorithm to simulate a hard-wired EW signal preprocessor is developed. An application oriented high level language version of the algorithm is presented.

A FORTRAN program which generates realistic data for signal analysis is described. Using the simulated data, an instruction level simulation program, and a language compiler, the algorithm is tested using an IBM 360 computer.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

I am deeply grateful to Applied Technology Division, Itek Corporation, Sunnyvale, California for the use of their facilities and the generous assistance and experience of their personnel, particularly the ATAC Support Software Group. I am particularly indebted to my wife, Phyllis, whose faith and devotion made this work possible and to whom it is dedicated.

I. INTRODUCTION

A. STATEMENT OF THE PROBLEM

Electronic Warfare Support Measures (ESM) is that division of Electronic Warfare (EW) concerned with exploitation of enemy use of the electromagnetic spectrum. ESM includes actions taken to search for, intercept, locate, and analyze electromagnetic radiations to support tactical military operations.

The continuously increasing density and complexity of today's electromagnetic environment, created by friendly as well as hostile forces, has fostered the requirement for more effective performance and greater sophistication in operational ESM systems. Current receiver technology has kept pace with this escalation, and signal intercept is virtually assured. However, the resulting increased data rates, existing and forecast manpower limitations, and limited reaction time available to the operational commander have imposed a requirement for faster and more accurate signal analysis, classification, and interpretation. Currently, major efforts are being put forth to achieve an effective ESM system design incorporating high speed digital computers for the purposes of automatic signal analysis and system control.

To perform effectively in present and future environments, an automated ESM system should exhibit the following characteristics:

1. Broad frequency range.
2. Simultaneous coverage of entire frequency range.
3. High probability of intercept.
4. Low false alarm rate.
5. Capability to recognize pulse or radar-like signals.
6. Capability to determine parameters associated with radar pulse trains.
7. Capability to classify signals according to type and source.
8. Direction finding capability.
9. Capability to handle data rates.
10. Capability to detect signals at ranges to radar horizon.

The successful design and development of a system exhibiting these characteristics rest upon three equally important elements: system hardware, system software, and the man-machine interface.

1. ESM Hardware/System Description

A discussion of ESM hardware as a specific problem area is beyond the scope of the thesis. A system concept compatible with the state of the art approach is presented here as a basic foundation for this study and future efforts in the ESM arena.

Conceptually, an airborne ESM system with the characteristics previously described could be implemented utilizing a combination of Instantaneous Frequency Measurement (IFM) and Superhet receivers. The IFM receivers, operating in conjunction with a high-gain pencil beam antenna, would cover the frequency bands of interest and function as the primary acquisition receivers of the system. In addition to contributing to increased system sensitivity, the high-gain pencil

beam antenna functions to reduce the instantaneous data rate and, in conjunction with azimuth centroiding, provides an accurate means of determining signal direction of arrival.

Additional data rate reduction is accomplished by routing each IFM output through a Pulse Train Separator (PTS), which functions as a signal preprocessor to separate stable pulse trains from the received data. Parameter measurements are performed on those stable trains separated and a computer word is generated to tag the associated emitter. The required capacity of the PTS in number of pulse trains separated is considered paramount to system operational effectiveness.

A computer word is generated for each pulse associated with non-stable pulse trains and those stable pulse trains exceeding the capacity of the PTS. These data are routed through a residual channel of the PTS to a high-speed digital computer for deinterleaving and parameter measurement. The parameters associated with each signal are then passed to a main frame computer for classification, threat analysis, and tactical environment update.

The IFM system employs an omni antenna for side lobe suppression to prevent signal poke-through in the receiving antenna minor lobes. The IFM sensitivity and antenna gain provide a capability for radar detection out to the radar horizon for aircraft altitudes in the neighborhood of 30,000 feet.

Priority signals, unclassifiable signals, and signals too complex for automatic analysis are automatically assigned

to digitally controlled superhet receivers for refined analysis. The superhet receivers cover the frequency bands of interest and operate in conjunction with an omni or low-gain broad beam antenna.

2. Man-Machine Interface

The human operator element of an automated ESM system, though normally an area addressed by human factors engineers, impacts heavily on the software requirements and will be discussed here in that regard.

Automation does not eliminate the requirement for human operators, but it does significantly affect the role the operator plays in the ESM system and it may seriously retard the operator's proficiency and ability to function efficiently as a back up in the event of a system failure. Thus, in deciding which system functions should be automated and which should be assigned to the human operator, it is imperative that the operator be assigned those functions that maximize the use of his capability to exercise judgment, improvise, reason inductively, and profit from experience.

The human operator must be able to control and direct the operation of the system. To do so, he must be provided relevant information by the system at the proper time, in the proper format, and on the proper displays. To obtain efficient and reliable software, these requirements must be clearly and accurately defined in the planning stage of the ESM system development.

3. ESM Software

It is felt that software represents the most troublesome problem in the design and development of an automatic ESM system. Assuming cost and risk vary directly with complexity, ESM software will account for a substantial percentage of total system costs and be the riskiest part of the system development. Since system performance hinges on software performance, implementation of an effective software test plan would serve a threefold purpose of cost reduction, risk reduction, and increased system performance.

The ability to successfully implement such a test plan requires a detailed description of the systems functions and a software modular building process. Each software module or set of modules is then integrated into the system and tested against sets or subsets of system functions according to prescribed test plans.

B. SCOPE OF THE THESIS

Simulation, via computer software, is one technique that is useful in software development and verification or testing. Realistic data, created by successful simulation of the ESM system hardware operating in a dynamic environment, utilized as inputs to a high-speed digital computer would provide the framework for implementation, testing, and optimization of ESM signal processing algorithms. The inherent ability to control the density of the electromagnetic environment through simulation would provide the capability to analyze worst case

situations, thereby enabling system designers to estimate computer capacity and reserve capacity requirements as well as data rate reduction requirements.

Given these simulation capabilities, an instruction-level simulation of the target computer provides a powerful means of debugging software and algorithm testing for accuracy and speed of operation. Applied Technology Division of Itek Corporation, Sunnyvale, California, has provided the Naval Postgraduate School with the programming support system for their ATAC computer. Instruction level simulation is only a part of this software package which includes the compiler programs for Programming Language for ATAC (PL/ATAC), a high level language developed for ATAC applications programming. Acquisition of the ATAC Programming Support System (APSS) gives impetus to the utilization of simulation as a technique for ESM software development and testing at the Naval Post-graduate School.

Total hardware simulation of the ESM system previously described is beyond the scope of the thesis. Total system simulation does, however, lend itself to a software "build" process and as a first effort of the process, three specific objectives of this work are as follows:

1. Provide a digital simulation of an IFM receiver operating in conjunction with a high-gain, pencil beam DF antenna in a dynamic radar environment.
2. Provide a digital simulation of a hard-wiredpulse train deinterleaver fed by an IFM receiver.
3. Describe and demonstrate the capabilities of APSS, particularly, the PL/ATAC programming language.

The overall objective of this work is to provide a building block as a basis for follow on study of ESM system software at the Naval Postgraduate School.

II. COMPUTER SYSTEM FOR ESM SOFTWARE DEVELOPMENT

A. GENERAL DESCRIPTION

APSS was written for Applied Technology's Airborne Computer (ATAC). ATAC, specifically designed for Electronic Warfare applications, is well suited for ESM software design, development and testing. A number of features of this 16-bit, fixed point, two's complement computer permit the establishment of a meaningful baseline for determining the relative merit of algorithms applicable to computer based ESM systems. Special features of interest are:

1. 437 nanosecond major cycle time.
2. Sixteen general registers which may be used as accumulators or as index registers.
3. Microprogrammable processor, permitting optimization of the instruction set for specific applications.
4. Memory types can be mixed in any combination of low power Read Only Memory (ROM), fast bipolar Random Access Memory (RAM), and slower core memory devices.
5. Memory expandable to 65K words.
6. Four basic instruction addressing modes: register, immediate, direct, and direct indexed.
7. Three supplemental addressing modes: register indexed, immediate short, and direct register.
8. Provisions for up to 16 Direct Memory Access (DMA) channels.
9. Provisions for up to four simultaneously active DMA channels.
10. Twelve hardware priority interrupts.
11. Modular, compact construction.

Additional detailed information, including the instruction set, may be obtained from ATAC, Applied Technology Airborne Computer, Vol. I, Principles of Operation.

B. ATAC PROGRAMMING SUPPORT SYSTEM (APSS)

This software support system allows for the preparation and debugging of programs via an IBM 360/370 computer system. APSS, through simulation of the ATAC computer and associated peripheral devices, permits the applications programmer to optimize programs for execution speed and memory utilization. The APSS system is written in a combination of PL/1, FORTRAN IV, and IBM assembler language and includes the programs listed below.

- APSS Monitor
- ATAC Instruction Assembler
- ATAC Instruction-Level Simulator
- Relocating and Linking Loader
- Microcode Assembler
- Microcode-Level Simulator
- PL/ATAC Compiler
- Library Control System
- Subroutine Library

Arrangements were made with Applied Technology to use portions of APSS and consequently, the ATAC Instruction-Level Simulation Program and the PL/ATAC Compiler are currently operational on the IBM 360/67 at the Naval Postgraduate School. Information concerning the implementation of these programs and instructions pertaining to the use of the programs are contained in Appendix A. Additional APSS information is available in ATAC, Applied Technology Airborne Computer, Volume 2, ATAC Programming Support System (APSS). The documents in this volume provide a detailed description of how to set up and use the APSS system. Those documents applicable to this work are:

1. Simulation of instruction execution.
2. Tracing option of the 16 general registers. The contents of each register can be printed after the execution of an instruction.
3. Accumulated program execution time printed with each line of trace.
4. Instruction execution histograms by frequency of execution or time of execution can be created and printed.
5. Partial and total simulated memory dump options.
6. Initialization of program data arrays allowing data to be read into virtual program memory prior to program simulation.

The APSS Instruction-Level simulator program includes a large repertoire of error messages to assist the applications programmer in debugging job management and ATAC instruction usage errors.

C. PL/ATAC

PL/ATAC, Programming Language for ATAC, is a high level, block structured language designed specifically for the ATAC computer. The programming structure and numerous capabilities of the language are demonstrated in Appendix C via the Pulse Train Separator simulation program written in PL/ATAC. In addition to hardware simulation, a specific objective in writing this program was to provide programming examples of the PL/ATAC language to assist the interested programmer in learning PL/ATAC. The PL/ATAC manual (Volume 2, Section II, ATAC) is an excellent reference but may present some difficulty if used as a learning text.

PL/ATAC is a powerful language providing the programmer with programming control approaching that of assembly language

without compromising the high level of the compiler language. Additionally, ATAC assembly language may be intermixed with PL/ATAC increasing programming flexibility and control with no serious degradation of program readability.

An analysis to determine the effectiveness of applications programming in the PL/ATAC language versus ATAC assembly language was conducted by the author utilizing an assembler optimized ATAC benchmark program. The results of this analysis indicated a memory utilization increase by a factor of approximately 2.3 times the ATAC assembly language requirement and an execution time increase by approximately 2.4 times the ATAC assembly language execution time.

The PL/ATAC programmer loses some flexibility, as compared to the assembly language programmer, due to the limitation in the number of registers available (a maximum of five) to him. Also, the PL/ATAC compiler currently installed at the Naval Postgraduate School does not have a double precision capability. Double precision calculations, if required, must be accomplished in assembly language resulting in increased core space and execution time required.

III. ESM ENVIRONMENT/IFM SIMULATION

Investigation revealed the existence of a computer program [Ref. 1] designed to simulate an IFM receiver operating in conjunction with an omni antenna in a dynamic radar environment. This program was modified to incorporate a rotating DF antenna system. Routines were added to generate unstable pulse trains in the form of pulse 'jitter' and/or 'stagger'. Documentation was added to facilitate program readability.

The DATA Simulation Program generates simulated pulsed radar signal level data arriving at an IFM receiver via a rotating antenna system. Current capacity allows for simulation of up to ten emitter platforms with up to five emitters per platform. This capacity may be increased by increasing the array dimensions of the parameters associated with each emitter-platform combination. Emitter platform and IFM platform initial position information in X-Y grid coordinates, course in degrees clockwise from North, and speed in knots are set by the program input data.

Emitter characteristics, set by the program input data, consist of transmitter frequency in MHz, pulse repetition interval (PRI) in milliseconds, pulse width (PW) in microseconds, transmitter power in kilowatts, antenna gain in db, antenna beamwidth in degrees, antenna scan rate in seconds per revolution, and pulse jitter and/or stagger characteristics. A sin(x)/x antenna radiation pattern is assumed for all emitters.

IFM receiver characteristics, set by the program input data, consist of receiver threshold sensitivity in dbm, upper and lower frequency limits of the passband in MHz, minimum time measurement capability in microseconds, antenna gain in db, antenna scan rate in seconds per revolution, and antenna beamwidth in degrees.

Additional program inputs include simulation start and stop times in seconds, frequency bin size of the IFM digital processor, and random number generator calling arguments. Complete input data description is depicted in the program listing following the description of the program variables. A complete program listing is contained in Appendix B.

Utilizing the input data described, the Data Simulation Program computes the position, course, speed, and range of each emitter platform relative to the IFM platform. The closest point of approach (CPA) and distance to the CPA is computed for each emitter platform to determine if and when a platform will be within the range of detection of the IFM. Those emitters out of the detection range during the simulation are listed as such. The IFM antenna's and all emitter antennas' intial pointing directions are set by a random process utilizing a uniform random number generator. The time at which each emitter platform enters the capture area of the IFM antenna's main beam is calculated and used as a basis for computing a random transmit time for the emitters associated with that platform.

Emitter maximum effective radiated power (P_E) is then computed utilizing the mathematical model given by

$$P_E = P_t \, G_t \, \lambda^2/16\pi^2$$

where

P_t = Transmitted Power,

G_t = Emitter Antenna Gain,

and λ = Wavelength of Signals.

Emitter maximum range of detection (R_m) is computed by:

$$R_m = (P_E \, G_r/T_h)^{\frac{1}{2}}$$

where

G_r = IFM Antenna Gain,

and T_h = IFM Receiver Threshold Sensitivity.

For each emitter within detection range, the minimum antenna gain (S_{min}) required to break the IFM threshold sensitivity is then computed by

$$S_{min} = T_h/G_r \, P_r \, r^2$$

where

r = Distance to the Emitter.

Using the value of emitter transmit time previously calculated, the emitter antenna pointing direction is calculated. Emitter antenna gain as a function of the relative angle between the emitter antenna and the IFM antenna is then calculated and compared with the minimum antenna gain required to break the IFM threshold sensitivity. If the signal level is sufficient to break the IFM threshold, the time of arrival (TOA) of the pulse at the IFM receiver is calculated.

If the pulse does not break the IFM threshold, the emitter transmit time is incremented by the emitter PRI, the antenna position is updated, and the antenna gain calculation process is repeated until a signal level of sufficient strength is transmitted. Should no pulse be received in the scan time of one beamwidth of the IFM antenna, the subsequent scan of the receiver antenna is checked and the process continues until a pulse is received or until simulation stop time.

The time of arrival of the first pulse from each emitter is stored in an array and, upon completion of this process for all emitters, the array is searched for the lowest time of arrival. The receiver antenna bearing at this TOA is calculated and provided as output data along with the parameters of the associated emitter. The emitter platform position is updated and the time of arrival of the next pulse from this emitter is calculated and stored in the array. The array is then searched for the next lowest time of arrival and the process repeated until simulation stop time.

The output of the Data Simulation Program consists of parameters associated with each pulse received by the IFM receiver in the proper time sequence. These parameters include the frequency, pulse width, bearing and time of arrival.

This data, as prepared for the ATAC simulation programs, consists of five integers for each pulse. These integers represent the following:

1. Emitter bearing. An integer in the range of 0-359. This word represents the IFM antenna bearing at the time of arrival of the pulse.
2. Frequency bin. The bins are numbered in ascending order of frequency from 1-128. This range may be altered as desired by the programmer.
3. Pulse width. The pulse width word is an integer in the range 0-65,535. The time scaling factor is arbitrary. Current convention is that the least significant bit of the pulse width word corresponds to .1 microseconds. This allows for representation of pulse widths from 0-6.5 milliseconds to within 0.05 microseconds.
4. Time of arrival. Time of arrival is expressed as a double precision (32 bit) integer. Word 4 is the least significant half of this integer and holds time values up to 0.84 seconds with the least significant bit corresponding to 25.6 microseconds.

5. Time of arrival. The upper half of the time word holds time values up to 15.3 hours.

The Data Simulation Program produces these outputs in printed format and in a punched card format suitable for acceptance by the ATAC simulation program. A sample of the program output is contained in Appendix B following the program listing. The program flow chart is illustrated in Figures 1 and 1a.

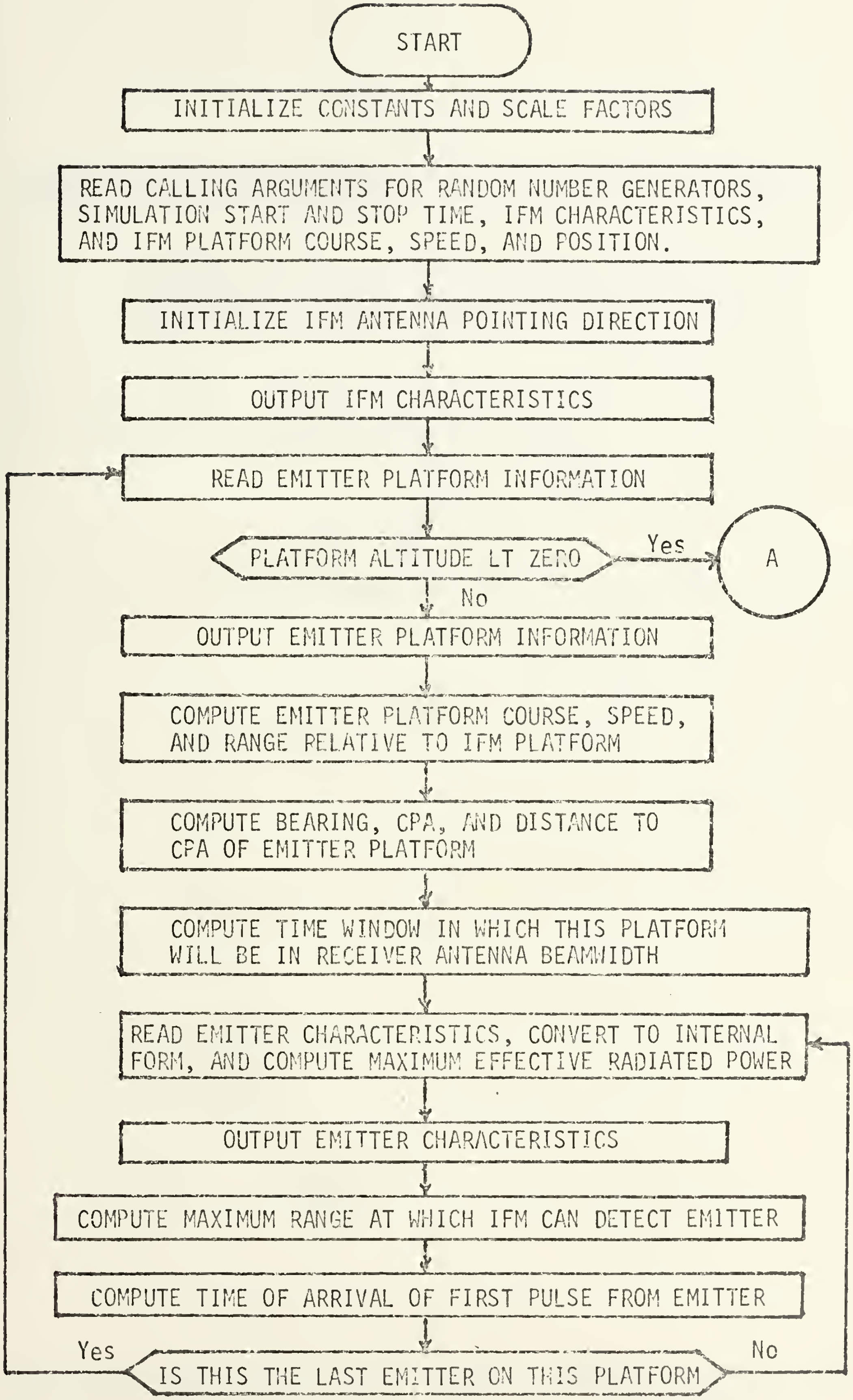

FIGURE 1. Data Simulation Program (Simplified Flowchart)

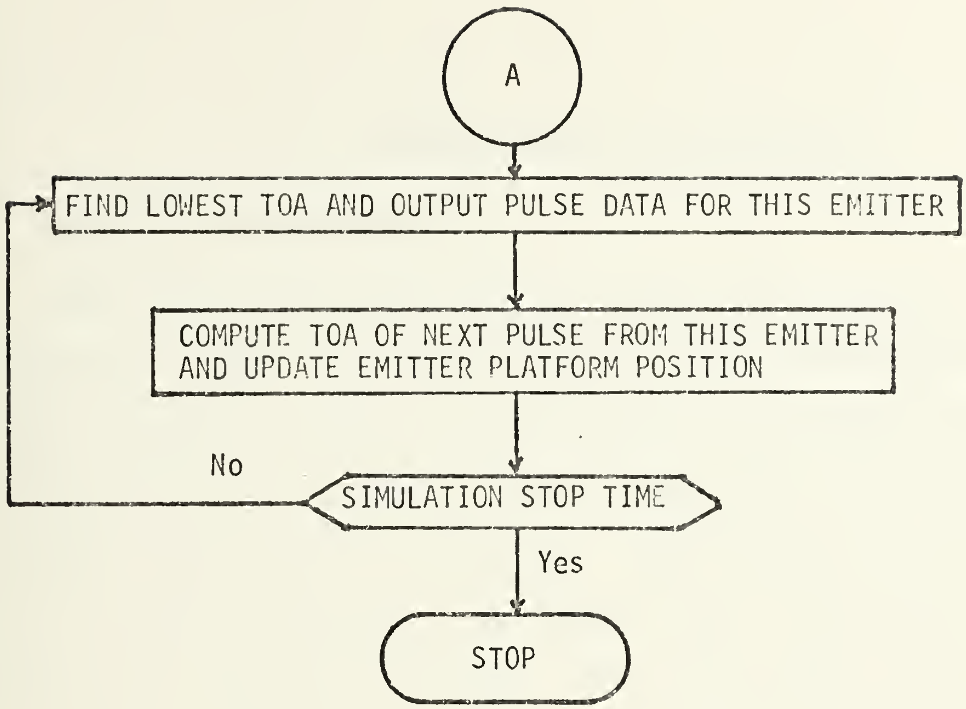

FIGURE 1a. Data Simulation Program (Simplified Flowchart)

IV. THE PULSE TRAIN SEPARATOR

A. CONCEPT OF OPERATION

Data received by the IFM are routed to the Pulse Train Separator (PTS). If a pulse train has a stable pulse repetition frequency (PRF) the PTS locks on the train and tracks the train for a period of time determined by the receiver antenna scan rate and beamwidth. The signal's pulse width (PS) and pulse repetition interval (PRI) are measured and digitized along with azimuth, time first seen and frequency. These five parameters are used to tag the associated emitter and are routed to the main frame computer for emitter classification. Direction of arrival is computed by azimuth centroiding based on the receiver antenna bearing at the time of arrival of the first and last pulse of the signal. The possibility exists that, due to sin(x)/x characteristics of emitter antennas, centroiding will be performed on ragged signal inputs. This is not considered degrading to system performance as meaningful direction of arrival information will still result.

An inherent characteristic of the Pulse Train Separator is that those stable trains with the lower values of PRI will be separated first. This feature of the PTS is also desirable as the data rate reduction role of the device is enhanced. The required capacity of the device in this role of data rate reduction has not been established and is considered a most

important factor in the overall ESM system design. Failure to provide the proper level of preprocessing could result in system software and hardware overloading and, consequently, degraded system performance.

Those stable signals exceeding the capacity of the Pulse Train Separator and non-stable signals such as frequency agile signals, jittered pulse trains, staggered pulse trains, and other exotic signals are routed through a residual signal channel of the pulse train separator to the high speed mini computer for parameter measurement. For each of these pulses received, a computer word will be generated for frequency, time of arrival, pulse width, and direction of arrival. Pulse deinterleaving and such parameter measurements as pulse repetition interval and pulse jitter and stagger characteristics will be performed by the mini computer for emitter classification.

B. PTS SIMULATION PROGRAM

The Pulse Train Separator Simulation Program is designed to simulate a hardware preprocessing system with a capability of separating up to a maximum of ten stable pulse trains from the output of an IFM receiver.

Input data to the program consist of a series of five integer words for each pulse generated by the Data Simulation Program; direction of arrival in degrees clockwise from North, frequency by bin number, double precision time of arrival (two words) in seconds with the least significant bit

corresponding to 25.6 microseconds and pulse width in seconds with the least significant bit corresponding to .1 microseconds. These data are loaded into a large input buffer prior to the start of the program simulation.

Each pulse within the buffer is compared in frequency with previously received pulses within the same receiver antenna beamwidth. If no match in frequency occurs the pulse is considered a new signal and a pointer address is stored in a New Signal Pointer File. If a frequency match occurs, the pulse width of the signals are compared. Upon obtaining a match in pulse width the matched pulse's pointer address is stored in a five element array within the New Signal Pointer File. Upon receipt of five pulses of a pulse train, the program calls a local re-entrant procedure (PRICOMP) wherein the stability of the pulse train is determined. For purposes of this simulation a pulse train is considered unstable if the variation in time interval between pulses is greater than 51.2 microseconds and/or three or more missing pulses are encountered in ragged signals. Upon determining a pulse train to be stable, the pulse repetition interval of the signal is calculated and the signal is flagged as a stable train. The number of available channels in the Pulse Train Separator is then reduced by one and the signal is zeroed in the input buffer as each additional pulse is received. Parameters associated with this pulse train are stored and the signal bearing is updated as each pulse is received through incrementation of the input buffer.

Upon receipt of the last pulse expected from the pulse train, the centroid direction of arrival is calculated and the signal information is zeroed in the New Signal Pointer File as well as in the first five input buffer lcoations. The time of arrival of the last pulse expected is computed utilizing the scan time for one receiver antenna bandwidth.

Non-stable signals are flagged as such and remain in the input buffer. Upon completion of incrementation through the input buffer, all stable pulses, except those from trains that remain due to PTS capacity having been exceeded, have been zeroed from the input buffer. Parameters associated with these trains are stored in sequence according to pulse repetition interval.

The pulse information remaining in the input buffer, associated with the unstable pulse trains and with pulse trains exceeding the PTS capacity, is then compressed within the input buffer, thus allowing for simulation of the mini computer signal processing algorithm. In the initial design concept of this simulation program, these remaining data, representing the Pulse Train Separator residual channel, would be processed by an ATAC assembly language program incorporated as a local procedure within the PTS program using the code selection head as described in the PL/ATAC manual.

A complete PL/ATAC program listing of the Pulse Train Separator is contained in Appendix C. An assembly language

program generated by the PL/ATAC compiler is contained in Appendix D. A program flowchart is illustrated in Figures 2 and 2a.

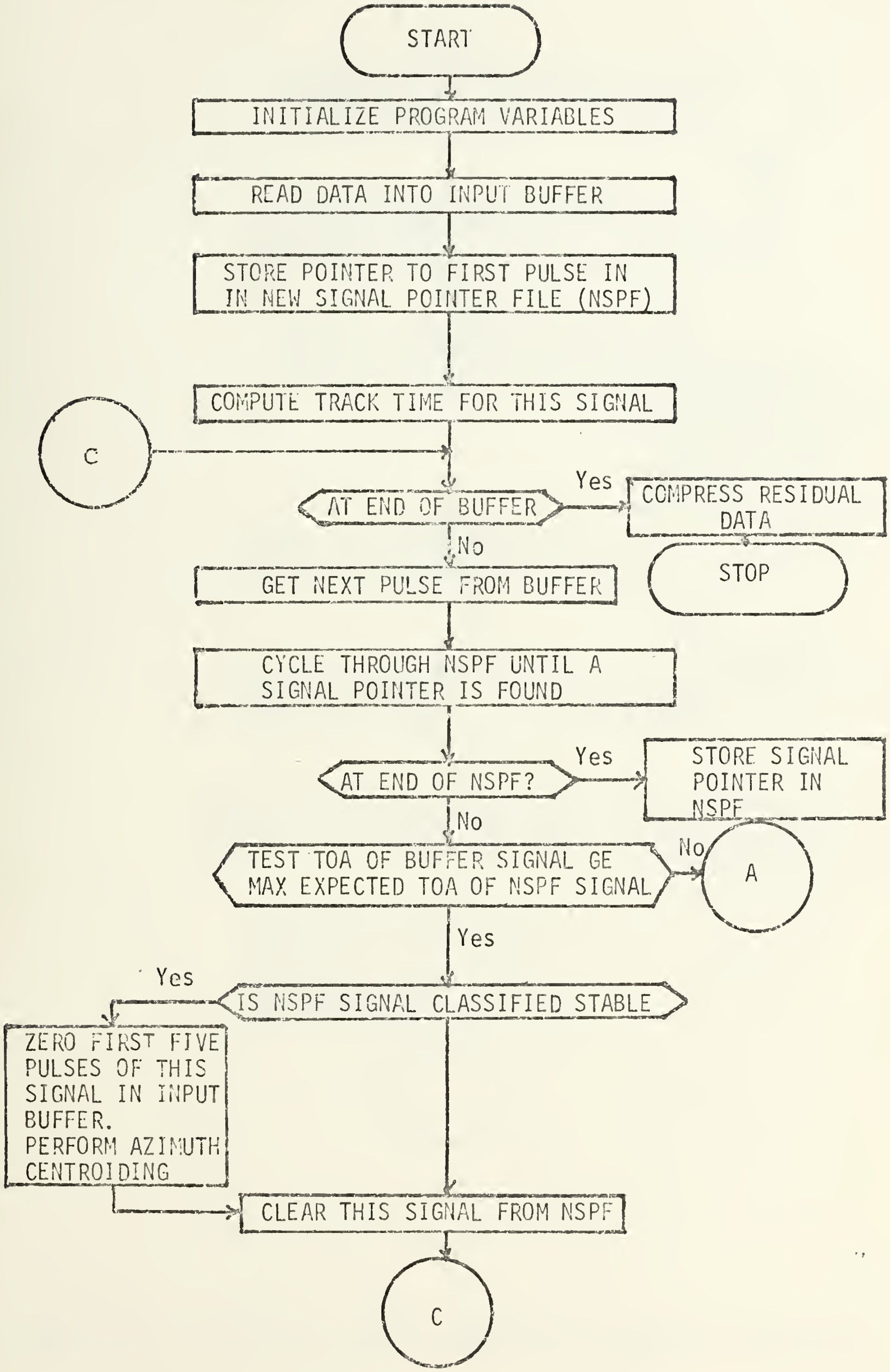

FIGURE 2. The Pulse Train Separator Program Flowchart

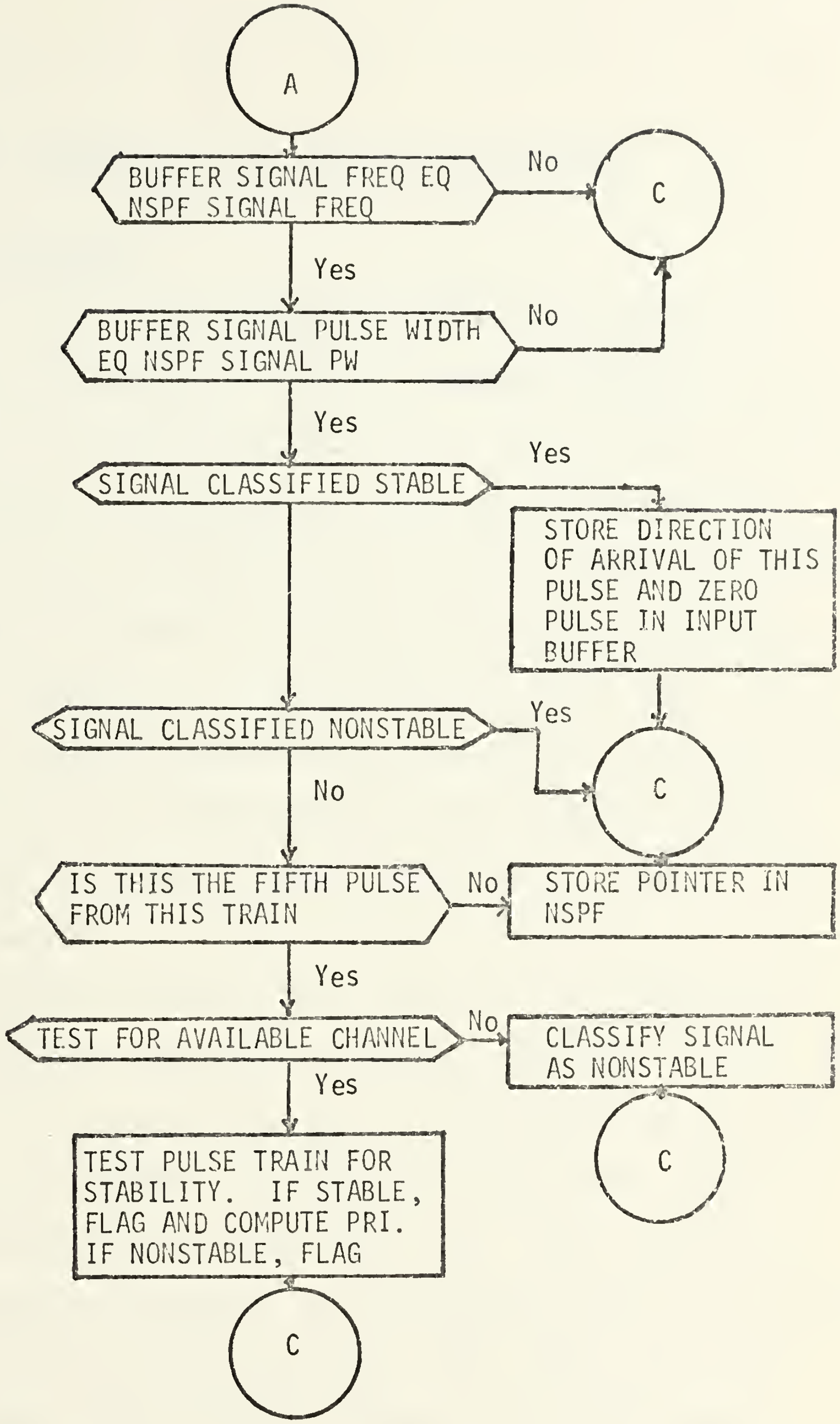

FIGURE 2a. The Pulse Train Separator Program Flowchart

V. DISCUSSION OF SIMULATION RESULTS

A. SUMMARY OF RESULTS

The Pulse Train Separator as described in Chapter IV was simulated utilizing the ATAC Programming Support System and the PL/ATAC Programming Language. The particular hardware system simulated was arbitrarily given the capability of deinterleaving and preprocessing up to nine radar pulse trains.

Input data, provided by the Data Simulation Program, consisted of a mixture of stable and non-stable pulse trains from eighteen emitters located on six platforms geographically positioned so as to be intercepted in one receiver antenna beamwidth. The receiver antenna beamwidth was arbitrarily set at five degrees.

Using a receiver antenna scan rate of two seconds per revolution, the Data Simulation Program generated a total of 365 pulses as IFM output during one antenna main beam sweep of the eighteen emitters. These pulses represent approximately 28 milliseconds of real time data. The Pulse Train Separator Simulation Program, in processing nine of the pulse trains, removed 277 pulses from the input data and routed 88 pulses through the residual channel. These 88 pulses represent the unstable pulse trains and the lower pulse repetition frequency (PRF) signals. Accurate pulse repetition interval calculations as well as accurate azimuth centroiding was performed on all preprocessed pulse trains.

B. CONCLUSIONS

The Data Simulation Program provides realistic data to the APSS programs. The program produces reasonably complex data in the form of jittered pulse trains and staggered pulse trains. Pulse trains transmitted at sufficiently low power levels are produced with missing pulses due to sin(x)/x effects.

The Pulse Train Separator Simulation Program demonstrated the use of the PL/ATAC language and functions as designed to preprocess ESM signals. This program is somewhat inefficient in its present form for two reasons. First, the limitation of five registers allocated to the programmer by the PL/ATAC compiler leads to a significant increase in program execution time as compared to a program written in ATAC assembly language. As the programmer gains experience with the PL/ATAC language, this limitation of the compiler may be overcome somewhat by programmer 'tricks' to gain use of registers.

The lack of a double precision capability in the current version of the PL/ATAC compiler installed on the IBM 360/67 at the Naval Postgraduate School created a severe handicap in writing the Pulse Train Separator Simulation Program and also contributes to program inefficiency. Each double precision calculation performed requires a program segment of assembly language and the housekeeping involved results in increased program execution time.

Even with these limitations it is felt that the simulation programs presented in this work provide the framework for ESM

software development and applications programming at the Naval Postgraduate School.

C. RECOMMENDATIONS

Based upon the experience and results obtained from this study, recommendations for follow-on study in the area of ESM software development are as follows:

1. Continue study of the Pulse Train Separator Program and optimize the basic algorithm.
2. Implement the resulting improved algorithm in ATAC assembly language.
3. Use the Data Simulation Program and optimized hardware simulation with APSS to develop and optimize ESM signal processing algorithms.
4. Continue use of the PL/ATAC compiler to develop and test hardware simulation and other programs prior to implementation in ATAC assembly language.

APPENDIX A

INSTRUCTIONS FOR INSTALLATION AND USE OF THE ATAC PROGRAMMING SUPPORT SYSTEM (APSS) ON THE IBM 360/67 OPERATING SYSTEM AT THE NAVAL POSTGRADUATE SCHOOL

A. INSTALLATION

The ATAC Programming Support System is currently loaded on the 2321 Data Cell. Data set names and their locations are as follows:

1. S1615.RDS.LOADLIB - CEL 007
2. S1616.RDS.APSS - CEL 005
3. S1615.RDS.PLATAC - CEL 001

In the event of failure of the Data Cell these data sets may be reloaded from the APSS source tape stored at the W.R. Church Computer Center under the name 'ATI.006'.

File #1 of this tape contains a description of the tape. File #2 contains examples of the JCL required to create and run APSS. The instructions contained in this file are superseded by the instructions contained here. The following JCL is required to list these files:

```
//(STANDARD JOB CARD)
//EXEC PGM-IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT2 DD SYSOUT=A, DCB=(LRECL=80,RECFM=F,BLKSIZE=80)
         LABEL=(1,SL,,IN),DCB=(BLKSIZE=800,LRECL=80,
         RECFM=FB),VOL=SER=ATI006,DSN=FILE1
// EXEC PGM=IEBGENER
// SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT2 DD SYSOUT=A,DCB=(LRECL=89,RECFM=F,BLKSIZE=80)
//SYSUT1 DD UNIT=(2400,,DEFER),DISP=(OLD,PASS),
         LABEL=(1,SL,,IN),DCB=(BLKSIZE=800,LRECL=80),
         RECFM=FB,VOL=SER=ATI006,DSN=FILE2
```

File #3 of the source tape contains the object module library of APSS and is required to build the APSS Instruction Level Simulator. File #4 contains the PL/ATAC Compiler object module. The following steps are required to build APSS from files 3 and 4.

1. Read files 3 and 4 from tape to disc storage. Special arrangements should be made to insure that the disc pack used has sufficient unallocated space for this step and the following step of this procedure. SPOOL3 has been made available in the past and is the disc used in the JCL of this example. The load modules from file 3 are moved into a disc file named 'S1614.RDS.LOADLIB'. File 4 is moved into a disc file named 'S1615.RDS.PLATAC'. The utility control statements may be changed to rename the files as they are being loaded onto the disc. The JCL statements required are as follows:

```
//(STANDARD JOB CARD)
//STEP1 EXEC PGM=IEHMOVE, REGION=100K
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=SYSDA,SPACE=(TRK,(40),,CONTIG)
//DA1   DD UNIT=2314,DSN=S1615.RDS.LOADLIB,
//SPACE=(TRK,(50,10,10),,CONTIG),
//DISP=(NEW,KEEP),VOL=SER=SPOOL3
//T1TAPE DD UNIT=(2400,,DEFER),DISP=(NEW,PASS),
         LABEL=(3,SL,,IN),
//DCB=(DEN=2,BLKSIZE=800,LRECL=80,RECFM=FB),
         VOL=SER=ATI006
//SYSIN DD *
COPY PDS=ATI.APSS.LOADLIB,TO=2314=SPOOL3,TODD=DA1,
         FROMDD=T1TAPE,     *
FROM=2400=(ATI006,3),RENAME=S1615.RDS.LOADLIB
COPY DSNAME=ATI.APSS.PLATAC,TO=2314=SPOOL3,
         FROMDD=T1TAPE,TODD=DA2,    *
FROM=2400=(ATI006,4),RENAME=S1615.RDS.PLATAC
//DA2 DD UNIT=2314,DSN=S1615.RDS.PLATAC,SPACE=(TRK,
         (20,10),RLSE),
//DISP=(NEW,KEEP),VOL=SER=SPOOL3,LABEL=RETPD=360
/*
```

2. The Library (named 'S1615.RDS.LOADLIB') now on the disc must be built into an absolute load module of the entire APSS system. The module thus built (named 'S1615.RDS.APSS') will be stored permanently on cell 005 of the data cell. This file may also be renamed as desired. The following JCL statements are required.

```
//(STANDARD JOB CARD)
//BUILD EXEC PGM=IEWL,REGION=150K,
//PARM='OVLY,XREF,LET,LIST,SIZE=(256K,20480)'
//SYSPRINT DD SYSOUT=A
//LIBRARY DD DSN=S1615.RDS.LOADLIB,UNIT=2314,
          VOL=SER=SPOOL3,DISP=SHR
//SYSLIB DD DSNAME=SYS1.FORTLIB,DISP=SHR
//SYSLMOD DD DSNAME=S1615.RDS.APSS,UNIT=2321,
          VOL=SER=CEL005,
//DISP=(NEW,KEEP),LABEL=RETPD=360,
//SPACE=(TRK,(300,20,2),RLSE)
//SYSUT1 DD UNIT=SYSDA,SPACE=(TRK,(19,19),,CONTIG),
          SEP=SYSLMOD
//SYSLIM DD*
INCLUDE LIBRARYEPMUDLN
CHANGE MSIMEIHESAPDN
INCLUDE LIBRARY (APSSMON)
INCLUDE LIBRARY (MSIM4,ASEM5,SIM16A,SIMTR1,SIMI01,GUL)
INCLUDE LIBRARYEXPLMONN
OVERLAY A1
INSERT MSIMUL,*MSIMULA,IHENTRY,IHESAP
INSERT MINT, IN, OUT
INSERT IHEDBN, IHEXID
INSERT IHEBSM, IHECSM
INSERT IHEBSK, IHEIOX, IHEIOP, IHEDID, IHEDOB
INSERT IHEBIB, IHEDOA, IHEIOB
INSERT IHEIOA, IHEOCL
INSERT IHEBSD, IHEBSF
INSERT IHEJXS
INSERT IHEOSD, IHEOST, IHEBSI
INSERT IHEVPE, IHEDMA, IHEVFB
INSERT IHEDNC, IHEVFD, IHEVFA, IHEVPD, IHEVPB, IHEVSC
INSERT IHEVSD, IHEVFE, IHEDCN, IHEUPB
INSERT IHEVFC, IHEVPE, IHEVPG, IHEVQB, IHEVQC
INSERTIHEABN, IHEIOD, IHEIOF, IHEPRT, IHEVQA, IHESPRT
INSERT IHEBEG, IHEERR, IHESIZ
INSERT MISET
OVERLAY A1
INSERT ASSEM, REWIND, REW72, DSKOUT, CARDIN, DISKIN,
       ERPRT, PRIADD
```

```
INSERT WRDATA, PRICOM, PRINOR, WRITEX, REFTIT, PREF,
       ERTIT
OVERLAY A1
INSERT PARMRD, PRESIM
OVERLAY A1
INSERT SMLTR, NRMTRM, STRTSM, TPAGE
INSERT RDCRD, ABNRMT, ARTHFP, TRACE, HGRAM, HGRAMI,
       HGRAMS
INSERT IOINIT, ACTIVE, SIMTIM
INSERT DEVDTA, ACT, TIME, INT, RAND, DEADT, DEBUG
OVERLAY A2
INSERT LEVEL, DMAIOI, DMAIOA, DMAIOD, RIOIO
INSERT REMACT, DMA, DMATM, RIO, RIOTM, RIOINT, INTOLY,
       DMAINT
INSERT DTRAN, PUTACT, RANDOM
OVERLAY A1
INSERT HGPRNT
OVERLAY A1
INSERT LAODER
OVERLAY A1
INSERT LINK, ENTEXT, SLLH
OVERLAY A1
INSERT PLATAC, IOPACK
OVERLAY $OBJECTEREGIONN
INSERT OBJECT, INIT, LIB, RCALPH, RCHEX, RCINT
INSERT MDATE
OVERLAY $DUMPEREGIONN
INSERT SMDUMP,PAGE
ENTRY MAIN
NAME APSS
/*
```

3. The PL/ATAC compiler object module on the disc must be moved to the data cell.

```
//(STANDARD JOB CARD)
//EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DISP=SHR,UNIT=2314,VOL=SER=SPOOL3,
//DSN=S1615.RDS.PLATAC
//SYSUT2 DD DISP=(NEW,KEEP),UNIT=2321,VOL=SER=CEL001,
//DCB=(RECFM=FT,BLKSIZE=7188),LABEL=RETPD=360,
//SPACE=(7188,(20,0)),DSN=S1615.RDS.PLATAC
//SYSIN DD *
GENERATE
/*
```

4. The Library ('S1615.RDS.LOADLIB') must now be moved from the disc to the data cell for permanent storage. This library contains modules necessary for successful execution of the PL/ATAC compiler. The required JCL statements to accomplish this step are as follows:

```
//(STANDARD JOB CARD)
//EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DISP=SHR,UNIT=2314,VOL=SER=SPOOL3,
//DSN=S1615.RDS.LOADLIB
//SYSUT2 DD DISP=(NEW,KEEP),UNIT=2321,VOL=SER=CEL007,
//DSW=S1615.RDS.LOADLIB,SPACE=(7294,(61,0,14)),
//DCB=(RECFM=UT,BLKSIZE=7294),LABEL=RETPD=360
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(20,5))
//SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(20,5))
//SYSIN DD *
COPY OUTDD=SYSUT2,IWDD=SYSUT1
/*
```

B. USER INSTRUCTION

The JCL required to compile and execute PL/ATAC programs is as follows:

```
//(STANDARD JOB CARD)
//PLATAC EXEC PGM=XPLMON,PARM='ALTER',REGION=200K
//STEPLIB DD DSN=S1615.RDS.LOADLIB,UNIT=2321,DISP=SHR,
          VOL=SER=CEL007
//PROGRAM DD DSN=S1615.RDS.PLATAC,UNIT=2321,VOL=SER=CEL001,
          DISP=SHR
//SYSPRINT DD SYSOUT=A
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A
//FT27F001 DD DSN=&&PLCODE,SPACE=(TRK,(20,10),RLSE),
//    DCB=(BLKSIZE=3520,LRECL=80,RECFM=FB),DISP=(NEW,PASS,DELETE),
//    UNIT=SYSDA
//SYSIN DD *

          .
          .

     PL/ATAC Program

          .
          .
```

```
//APSS EXEC PGM=APSS REGION=200K
//STEPLIB DD DSN=S1615.RDS.APSS,UNIT=2321,DISP=(OLD,KEEP),
//   VOL=SER=CEL005,DCB=BLKSIZE=400
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3325)
//FT08F001 DD UNIT=SYSDA,SPACE=(CYL,1),DCB=BUFNO=1
//FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(7,2)),
                 DCB=(RECFM=VBS,BLKSIZE=7180,LRECL=92,BUFNO=1)
//FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(7,2)),
                 DCB=(RECFM=VBS,BLKSIZE=4204,LRECL=42,BUFNO=1)
//FT20F001 DD UNIT=SYSDA,SPACE=(CYL,(7,2)),
    DCB=(BUFNO=1,RECFM=VBS,BLKSIZE=2004,LRECL=500)
//FT27F001 DD DSN=&&PLCODE,UNIT=SYSDA,DISP=(OLD,DELETE),
    DCB=BUFNO=1
//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3325)
//SYSIN DD *
$END
/*
```

APPENDIX B

DATA SIMULATION PROGRAM FORTRAN LISTING AND OUTPUT

A description of program variables is included in the Data Simulation Program listing. The input data deck setup is also included in the listing.

```
C                    DATA SIMULATION PROGRAM (DSP)
C
C
C     THIS PROGRAM SIMULATES AN IFM RECEIVER OPERATING IN CONJUNCTION
C     WITH A HIGH-GAIN PENCIL BEAM ANTENNA IN A PULSE RADAR
C     ENVIRONMENT.  OUTPUT DATA CONSISTS OF EMITTER BEARING,
C     FREQUENCY, TIME OF ARRIVAL(DOUBLE PRECISION), AND PULSE WIDTH
C     FOR EACH PULSE GENERATED BY THE PROGRAM.
C
C
C                          DESCRIPTION OF VARIABLES
C
C
C     ANG(I,J)      ANTENNA TRUE POINTING ANGLE OF THE J EMITTER ON
C                   THE I PLATFORM
C     ANGLE         EMITTER ANTENNA ANGLE AT TRANSMIT TIME
C     ANGRF         RCVR ANTENNA BEARING AT TIME EMITTER PLATFORM
C                   ESCAPES RCVR ANTENNA BEAMWIDTH
C     ANGRI         RCVR ANTENNA BEARING AT TIME EMITTER ENTERS BW
C     ANT(I,J)      ANTENNA SCAN RATE OF THE J EMITTER ON THE I PLATFORM
C     BEAM          INPUT VALUE OF ANTENNA BEAM WIDTH
C     BIN           IFM FREQUENCY BIN SIZE
C     BNG(I)        TRUE BEARING OF THE I PLATFORM
C     BNGI          COMPUTATION VALUE OF BNG(I)
C     BW(I,J)       ANTENNA BEAM ATTENUATION FACTOR OF THE J EMITTER
C                   ON THE I PLATFORM
C     BWI           CALCULATION VALUE OF BW(I,J)
C     CON1          58.9*ANTENNA DIMENSION/WAVELENGTH
C     CPA           CLOSEST POINT OF APPROACH
C     CSEI          INPUT VALUE OF THE COURSE OF THE I PLATFORM
C     CSEO          INPUT VALUE OF THE COURSE OF THE IFM PLATFORM
C     D1            DISTANCE TO CPA
C     D2            DISTANCE TO MAX RANGE AT WHICH AN EMITTER CAN BE
C                   DETECTED
C     DEGR          CONVERSION FACTOR TO CONVERT DEGREES TO RADIANS
C     DEL           DELAY TIME
C     F(I,J)        FREQUENCY OF THE I EMITTER ON THE J PLATFORM
C     FMAX          MAX RECEIVE FREQUENCY OF THE IFM
C     FMIN          MINIMUM RECEIVE FREQUENCY OF THE IFM
C     FREQ          INPUT VALUE OF EMITTER FREQUENCY
C     FSCALE        CONVERSION FACTOR MEGA UNITS TO UNITS
C     GAIN          INPUT VALUE OF EMITTER ANTENNA GAIN
C     ICT           COUNTER FOR NUMBER OF PULSES RECEIVED BY IFM
C     IF(I,J)       SELECTOR FOR DESIRED TYPE OF FREQUENCY VARIATION
C     INEXT(L)      PLATFORM SUBSCRIPT OF THE SECOND OR MORE PULSES
C                   WHEN MORE THAN ONE PULSE IS RECEIVED
C                   SIMULTANEOUSLY BY THE IFM
C     ISTGR         CALCULATION INDEX FOR PULSE STAGGER
```

```
C       IPT(I,J)        SELECTOR FOR THE DESIRED TYPE OF OUTPUT POWER
C                       VARIATION
C       IPW(I,J)        SELECTOR FOR THE DESIRED TYPE OF PULSE WIDTH
C                       VARIATION
C       ISCAN(I,J)      SELECTOR FOR THE DESIRED TYPE OF SCAN RATE
C                       VARIATION
C       ITEMP           PLATFORM SUBSCRIPT OF THE CURRENT PULSE
C                       RECEIVED BY THE IFM
C       JNEXT(L)        EMITTER SUBSCRIPT USED FOR MULTIPLY RECEIVED PULSES
C       JTEMP           EMITTER SUBSCRIPT OF THE CURRENT PULSE RECEIVED
C                       BY THE IFM
C       N               BIN NUMBER OF THE FREQUENCY OF THE RECEIVED PULSE
C       NGAUS           INTERGER CALLING ARGUMENT FOR GAUS
C       NMTR(I)         NUMBER EMITTERS ON THE I PLATFORM
C       NOUT(I)         ARRAY TO PREPARE OUTPUT DATA FOR THE ATAC COMPUTER
C       NPW             INTEGER REPRESENTATION OF PULSE WIDTH
C       NTIME           INTEGER REPRESENTATION OF SIMULATION TIME
C       NTIME1          LOWER HALF OF DOUBLE PRECISION TIME
C       NTIME2          UPPER HALF OF DOUBLE PRECISION TIME
C       NUNI            INTEGER CALLING ARGUMENT FOR RANDU
C       PI              3.1415926
C       PI2             PI/2
C       NBIN            NUMBER OF FREQUENCY BINS IN THE IFM BANDWIDTH
C       PJT(I,J)        PER CENT PULSE JITTER
C       PJTI            INPUT VALUE OF PERCENT PULSE JITTER
C       POWER (I,J)     EFFECTIVE RADIATED POWER OF EMITTER
C       POWIJ           CALCULATION VALUE OF POWER(I,J)
C       POW             POWIJ WITH RANGE ATTENUATION
C       PRI(I,J)        PULSE REPETITION INTERVAL
C       PRIN            INPUT VALUE OF PRI
C       PSCALE          CONVERSION FACTOR FOR CONVERTING FROM MILLI
C                       UNITS TO UNITS
C       PTRANS          INPUT VALUE OF TRANSMITTER POWER
C       PULSE           CALCULATION VALUE OF PW(I,J)
C       PW(I,J)         PULSE WIDTH
C       R               RANGE
C       RANT            INITIAL BEARING OF RCVR ANTENNA
C       RATE            RATE OF ANTENNA ACAN
C       RBW             RCVR ANTENNA BEAMWIDTH AT HALF POWER POINTS
C       RCSE            DIRECTION OF RELATIVE MOTION
C       RCVBNG          RCVR ANTENNA BEARING AT TOA OF PULSE
C       RCVR            FREQUENCY OF THE RECEIVED PULSE
C       RCVSIG          SIGNAL LEVEL OF THE RECEIVED PULSE
C       RDEG            CONVERSION FACTOR RADIANS TO DEGREES
C       RMAX            MAXIMUM RANGE AT WHICH AN EMITTER CAN BE DETECTED
C       RRATE           RCVR ANTENNA SCAN RATE
C       SCNMIN          MINIMUM VALUE OF ANTENNA GAIN FOR WHICH AN EMITTER
C                       CAN BE DETECTED BY THE IFM
```

```
C      SEND(I,J)    TIME OF TRANSMISSION OF PJLSE FROM EMITTER
C      SIGFRQ(I,J)  FREQUENCY OF THE TRANSMITTED PULSE
C      SIGNAL       CALCULATION VALUE OF RCVSIG
C      CPDI         PLATFORM INPUT SPEED
C      SPDO         INPUT IFM PLATFORM SPEED
C      STGRI        INPUT VALUE OF STAGGER RATIO
C      STGR         CALCULATION VALUE OF STAGGER
C      TF           SIMULATION STOP TIME
C      TFIN(I,J)    TIME AT WHICH EMITTER ESCAPES RCVR ANTENNA BW
C      THATAI       BEARING OF THE CPA OF THE I PLATFORM
C      THRESH       THRESHOLD SENSITIVITY OF THE IFM
C      TINI         TIME EMITTER ENTERS RCVR ANTENNA BEAMWIDTH
C      TINE         MINIMUM TIME MEASUREMENT CAPABILITY OF THE IFM
C      TNEXT(I,J)   TOA OF NEXT PULSE TRANSMITTED BY  THE J EMITTER
C      TO           SIMULATION START TIME
C      TOF          PJLSE TIME OF FLIGHT FROM EMITTER TO IFM
C      TSCALE       CONVERSION FACTOR  KILO UNITS TO UNITS
C      TSKTS        CONVERSION FACTOR  KNOTS TO YARDS PER SECOND
C      TT           PJLSE TRANSMIT TIME
C      TWOPI        2*PI
C      U            ARGUMENT IN SIN(X)/X FUNCTION
C      VL           VELOCITY OF LIGHT
C      VTI          RELATIVE VELOCITY OF THE I PLATFORM
C      VX(I)        RELATIVE VELOCITY OF I PLATFORM IN X DIRECTION
C      VXI          CALCULATION VALUE OF VX(I)
C      VXO          VELOCITY OF IFM PLATFORM IN X DIRECTION
C      VY(I)        RELATIVE VELOCITY OF I PLATFORM IN Y DIRECTION
C      VYI          CALCULATION VALUE OF VY(I)
C      VYO          VELOCITY OF IFM PLATFORM IN Y DIRECTION
C      WSCALE       CONVERSION FACTOR  MICRO UNITS TO UNITS
C      X(I)         RELATIVE POSITION OF I PLATFORM IN X DIRECTION
C      XI           INPUT X POSITION OF THE I PLATFORM
C      XO           INPUT X POSITION OF IFM PLATFORM
C      Y(I)         RELATIVE POSITION OF I PLATFORM IN Y DIRECTION
C      YDS          CONVERSION FACTOR   NAUTICAL MILES TO YARDS
C      YI           INPUT Y POSITION OF THE I PLATFORM
C      YO           INPUT Y POSITION OF IFM PLATFORM
C      Z(I)         RELATIVE POSITION OF THE I PLATFORM IN THE Z DIRECTIO
C      ZI           INPUT Z POSITION OF THE I PLATFORM
C      ZO           INPUT Z POSITION OF THE IFM PLATFORM
C
C
```

```
C        INPUT DATA DECK STRUCTURE
C
C        INPUT CARD SEQUENCE
C
C        IFM PLATFORM INFORMATION              THREE CARDS
C        EMITTER PLATFORM INFORMATION          ONE CARD
C        EMITTER INFORMATION                   TWO CARDS
C                .
C                .
C                .
C        (UP TO FIVE EMITTERS)
C                .
C        EMITTER PLATFORM INFORMATION          ONE CARD
C        EMITTER INFORMATION                   TWO CARDS
C                .
C                .
C                .
C        (UP TO TEN PLATFORMS)
C                .
C        END OF DATA CARD
C
C        INPUT CARD FORMAT      IFM PLATFORM INFORMATION
C
C        CARD   COLUMNS   FORMAT   VARIABLE    UNITS
C
C          1      1-10     I10      NUNI         ODD INTEGER
C                11-20     I10      NGAUS        ODD INTEGER
C                21-30     F10.0    VL           10.EXP(6) YDS/SEC
C                31-40     F10.0    TO           SECONDS
C                41-50     F10.0    TF           SECONDS
C                51-60     F10.0    TINT         MICROSECONDS
C                61-70     F10.0    THRESH       DBM
C          2      1-10     F10.0    XO           NAUTICAL MILES
C                11-20     F10.0    YO           NAUTICAL MILES
C                21-30     F10.0    ZO           FEET
C                31-40     F10.0    CSEO         DEGREES CLOCKWISE FROM NORTH
C                41-50     F10.0    SPDO         KNOTS
C                51-60     F10.0    FMAX         MHZ
C                61-70     F10.0    FMIN         MHZ
C          3      1-10     I10      NBIN         INTEGER
C                11-20     F10.0    RBW          DEGREES
C                21-30     F10.0    RSPR         SECONDS
C                31-40     F10.0    RGAIN        DB
C
```

```
C       INPUT CARD FORMAT      EMITTER PLATFORM INFORMATION
C
C       CARD  COLUMNS  FORMAT  VARIABLE   UNITS
C
C         1     1-10     F10.0   XI           NAUTICAL MILES
C              11-20     F10.0   YI           NAUTICAL MILES
C              21-30     F10.0   ZI           FEET
C              31-40     F10.0   CSEI         DEGREES CLOCKWISE FROM NORTH
C              41-50     F10.0   SPDI         KNOTS
C              51        I1      NMTR(I)      INTEGER
C
C       INPUT CARD FORMAT      EMITTER INFORMATION
C
C       CARD  COLUMNS  FORMAT  VARIABLE   UNITS
C
C         1     1-10     F10.0   FREQ         MHZ
C              11-20     F10.0   PRI          MILLISECONDS
C              21-30     F10.0   PW           MICROSECONDS
C              31-40     F10.0   PTRANS       KILOWATTS
C              41-50     F10.0   BEAM         DEGREES
C              51-60     F10.0   GAIN         DB
C              61-70     F10.0   SCAN         SECONDS PER REVOLUTION
C         2     1-10     F10.6   PJTI         REAL NUMBER PERCENT
C              11-20     F10.6   STGRI        REAL NUMBER PERCENT
C
C       END OF DATA  FORMAT   NEGATIVE NUMBER IN CC 21-30
C
```

```
      COMMON X(10),Y(10),Z(10),VX(10),VY(10),BNG(10),F(10,5),TINT,VXO,VY
     1O,THRESH,PI,PI2,SCNMIN,PRIN,SIGFRQ(10,5),PRI(10,5),PW(10,5),POWER(
     210,5),BW(10,5),SEND(10,5),TNEXT(10,5),SIGNAL(10,5),ANT(10,5),ANG(1
     30,5),XO,YO,ZO,CSEO,SPDO,VL,TWOPI,RDEG,DEGR,TO,TF,FMAX,FMIN,BIN,TT,
     4IF(10,5),PJT(10,5),STGR(10,5),IPT(10,5),ISCAN(10,5),INEXT(10),JNEX
     5T(10),NUNI,NGAUS,NMTR(10),TFINI(10,5),RSPR,RRATE,RBW,RGAIN,NOUT(40
     6)
C
C
C     INITIALIZE CONSTANTS
C
C
      PI2 = ARSIN(1.0)
      PI = 2.*PI2
      TWOPI = 4.*PI2
      RDEG = 180./PI
      DEGR = PI/180.
      CON1 = PI*58.9*DEGR
      CON2 = 10.*ALOG10(4.*PI)
      TSKTS = 0.5626111
      YDS = 2025.4
      FSCALE = 1.E6
      TSCALE = 1.E3
      PSCALE = 1.E-3
      WSCALE = 1.E-6
      IIJ = 0
      ICT = 0
C
C
C     SET IFM INFORMATION
C
C
      READ (5,17) NUNI,NGAUS,VL,TO,TF,TINT,THRESH
      READ (5,18) XO,YO,ZO,CSEO,SPDO,FMAX,FMIN
      READ (5,21) NBIN,RBW,RSPR,RGAIN
      VL = VL*FSCALE
      TINT = TINT*WSCALE
      FMAX = FMAX*FSCALE
      FMIN = FMIN*FSCALE
      BIN = (FMAX-FMIN)/FLOAT(NBIN)
C
C
C     SET IFM INITIAL ANTENNA POSITION
C
C
      CALL RANDU (NUNI,IV,RAND)
      NUNI = IV
      RANT = TWOPI*RAND
```

```
      RRATE = TWOPI/RSPR
C
C
C     OUTPUT IFM INFORMATION
C
C
      WRITE (6,23) XO,YO,ZO,CSEO,SPDO,FMIN,FMAX,BIN,NBIN
      WRITE (6,24) THRESH
      WRITE (6,30) RGAIN
      WRITE (6,31) RSPR
      CSEO = CSEO*DEGR
      SPDO = SPDO*TSKTS
      XO = XO*YDS
      YO = YO*YDS
      ZO = ZO/3.
      VXO = SPDO*SIN(CSEO)
      VYO = SPDO*COS(CSEO)
      I = 1
C
C
C     GET EMITTER PLATFORM INFORMATION
C
C
    1 READ (5,19) XI,YI,ZI,CSEI,SPDI,NMTR(I)
C
C
C     IF ALL EMMITERS HAVE BEEN ENTERED GO SORT TOA'S
C
C
      IF (ZI.LT.0.) GO TO 10
      WRITE (6,25) I,XI,YI,ZI,CSEI,SPDI
C
C
C     COMPUTE PLATFORM RELATIVE COURSE AND SPEED
C
C
      CSEI = CSEI*DEGR
      SPDI = SPDI*TSKTS
      VXI = SPDI*SIN(CSEI)-VXO
      VYI = SPDI*COS(CSEI)-VYO
      XI = XI*YDS-XO
      YI = YI*YDS-YO
      Z(I) = ZI/3.
      VTI = SQRT(VXI**2+VYI**2)
      RCSE = ATAN2(VXI,VYI)
C
C
C     COMPUTE RANGE AND BEARING TO PLATFORM
```

```
C
C
      R = SQRT(XI**2+YI**2)
      BNGI = ATAN2(XI,YI)
C
C
C     COMPUTE PULSE TIME OF FLIGHT FOR THIS PLATFORM
C
C
      TOF = R/VL
      X(I) = XI
      Y(I) = YI
      VX(I) = VXI
      VY(I) = VYI
      BNG(I) = BNGI
C
C
C     COMPUTE CPA AND DISTANCE TO CPA
C
C
      THETAI = ABS(PI-BNGI+RCSE)
      IF (THETAI.GT.TWOPI) THETAI=THETAI-TWOPI
      IF (THETAI.GE.PI2) GO TO 2
      D1 = R*COS(THETAI)
      CPA = R*SIN(THETAI)
      GO TO 3
    2 CPA = R
    3 JJ = NMTR(I)
C
C
C     COMPUTE TIME WINDOW IN WHICH THIS PLATFORM WILL BE IN THE
C     RECEIVER ANTENNA BEAMWIDTH
C
C
      IF (BNGI.LE.0.) BNGI=BNGI+TWOPI
      IF (ABS(BNGI-RANT).LE.((RBW/2.)*DEGR)) GO TO 4
      ANGRI = BNGI-RBW/2.*DEGR-RANT
      IF (ANGRI.LT.0.) ANGRI=ANGRI+TWOPI
      ANGRF = BNGI+RBW/2.*DEGR-RANT
      IF (ANGRF.LT.0.) ANGRF=ANGRF+TWOPI
      TINI = ANGRI/RRATE+TO
      GO TO 5
    4 TINI = TO
    5 TFIN = ANGRF/RRATE+TO
C
C
C     GET EMITTER INFORMATION
C
```

```
C
C
      DO 9 J=1,JJ
      READ (5,18) FREQ,PRIN,PULSE,PTRANS,BEAM,GAIN,ANT(I,J)
      READ (5,20) PJTI,STGRI
C
C
C     SET PARAMETERS
C
C
      PJT(I,J) = PJTI
      STGR(I,J) = STGRI
      PTRANS = PTRANS*TSCALE
      F(I,J) = FREQ*FSCALE
      PRIN = PRIN*PSCALE
      PRI(I,J) = PRIN
      PULSE = PULSE*WSCALE
      PW(I,J) = PULSE
      TFINI(I,J) = TFIN
      BW(I,J) = CON1/BEAM
C
C
C     COMPUTE EFFECTIVE RADIATED POWER
C
C
      POWIJ = 10.*ALOG10(PTRANS)+GAIN-20.*ALOG10(FREQ)+58.3335
      POWER(I,J) = POWIJ
C
C
C     OUTPUT PARAMETERS OF THIS EMITTER
C
C
      WRITE (6,26) J,F(I,J),PRI(I,J),PW(I,J),PTRANS,GAIN,BEAM,POWER(I,J)
      WRITE (6,27) ANT(I,J)
C
C
C     COMPUTE MAX RANGE OF DETECTION
C
C
      RMAX = 10.**((POWIJ+RGAIN-THRESH)/20.)
C
C
C      CALCULATE THE TOA OF THE FIRST PULSE
C
C
      IF (RMAX.GE.CPA) GO TO 6
      TNEXT(I,J) = TF+1.
      WRITE (6,28)
```

```
      GO TO 9
    6 CALL RANDU (NUNI,IV,RAND)
      NUNI = IV
      ANG(I,J) = RAND*TWOPI
      CALL RANDU (NUNI,IV,RAND)
      NUNI = IV
C
C
C     IF EMITTER RANGE GREATER THAN MAX RANGE OF DETECTION GO COMPUTE
C     DELAY TIME
C     OTHERWISE GO COMPUTE A TRANSMIT TIME WITHIN THE EMITTER
C     PRI OF THE TIME EMITTER WILL BE IN THE RECEIVER ANTENNA BEAM
C
C
      IF (RMAX.LT.R) GO TO 7
      DEL = 0.
      R1 = R
      GO TO 8
    7 D2 = CPA*SIN(ARCOS(CPA/RMAX))
      DEL = (D1-D2)/VTI+(R-RMAX)/VL
      R1 = RMAX
    8 TT = TINI+PRIN*RAND-TOF-PULSE+DEL
      IF ((TT+TOF+DEL).LT.TINI) TT=TT+PRIN
C
C
C     COMPUTE RANGE ATTENUATION
C
C
      POW = POWIJ-20.*ALOG10(R1)
C
C
C     COMPUTE MINIMUM EMITTER ANTENNA GAIN REQUIRED FOR RECEPTION
C
C
      SCNMIN = THRESH-POW-RGAIN
C
C
C     COMPUTE RECEIVED SIGNAL POWER LEVEL
C     IF ANTENNA GAIN (COMPUTED IN SCAN) WAS BELOW THE REQUIRED VALUE
C     FUNCTION SCAN WILL HAVE INCREMENTED TIME AT THE EMITTER PRI
C     UNTIL EMITTER ANTENNA GAIN AT SUFFICIENT LEVEL OR TIME EXCEEDS
C     SIMULATION STOP TIME
C
C
      SIGNAL(I,J) = POW+SCAN(I,J)
C
C
C     COMPUTE TOA
```

```
C
C
      TNEXT(I,J) = TINT*AINT((TT+TOF)/TINT)
      SEND(I,J) = TT
      SIGFRQ(I,J) = F(I,J)
    9 CONTINUE
C
C
C
C     GO GET NEXT PLATFORM INFO
C
C
      I = I+1
      GO TO 1
C
C
C     AT THIS POINT ALL OF THE EMITTER PLATFORMS AND ALL EMITTERS
C     HAVE BEEN READ IN AND THE TOA OF THE FIRST PULSE OF EACH
C     EMITTER HAS BEEN COMPUTED
C     ROUTINE BELOW IS TO FIND THE TOA WITH THE LOWEST VALUE
C
C
   10 NTGT = I-1
      WRITE (6,29)
   11 TEMP = TF
      L = 0
      ITEMP = 0
C
C
      DO 14 I=1,NTGT
      JJ = NMTR(I)
C
C
      DO 13 J=1,JJ
      TTRY = TNEXT(I,J)
      IF (TEMP.LT.TTRY) GO TO 13
      IF (TEMP.EQ.TTRY) GO TO 12
      TEMP = TTRY
      ITEMP = I
      JTEMP = J
      GO TO 13
   12 L = L+1
      INEXT(L) = ITEMP
      JNEXT(L) = JTEMP
      ITEMP = I
      JTEMP = J
   13 CONTINUE
C
```

```
C
   14 CONTINUE
C
C
      IF (ITEMP.EQ.0) STOP
C
C
C     CONVERT PULSE DATA TO FORMAT REQUIRED BY THE ATAC COMPUTER
C     AND OUTPUT DATA ON PRINTER AND CARDS
C
C
   15 RCVSIG = SIGNAL(ITEMP,JTEMP)
      RCVR = SIGFRQ(ITEMP,JTEMP)
      N = (RCVR-FMIN)/BIN+1.
      ICT = ICT+1
      RPW = 1.E7*PW(ITEMP,JTEMP)
      NPW = RPW
      IF ((RPW-NPW).GT..5) NPW=NPW+1
      TNEXT1 = TNEXT(ITEMP,JTEMP)
      RCVBNG = AMOD((RANT+TNEXT1*TWOPI/RSPR),TWOPI)*RDEG
      IRXB = RCVBNG
      NTIME = TNEXT1*1.E7/256.
      NTIME1 = MOD(NTIME,65536)
      NTIME2 = NTIME/65536
      WRITE (6,22) IRXB,ICT,N,NTIME2,NTIME1,NPW,TNEXT1
      IF (TNEXT1.GT.TF) STOP
      NOUT(IIJ+1) = IRXB
      NOUT(IIJ+2) = N
      NOUT(IIJ+3) = NTIME1
      NOUT(IIJ+4) = NTIME2
      NOUT(IIJ+5) = NPW
      IIJ = IIJ+5
      IF (IIJ.LT.40) GO TO 16
      IIJ = 0
      WRITE(7,32)NOUT
   16 IF (ICT.GE.800 ) STOP
C
C
C     GO CALCULATE THE TOA OF THE NEXT PULSE FROM THIS EMITTER
C
C
      CALL UPDATE (ITEMP,JTEMP)
      IF (L.EQ.0) GO TO 11
      ITEMP = INEXT(L)
      JTEMP = JNEXT(L)
      L = L-1
      GO TO 15
C
```

```
C
   17 FORMAT (2I10,6E10.0)
   18 FORMAT (8E10.0)
   19 FORMAT (5E10.0,I1)
   20 FORMAT (2F10.6)
   21 FORMAT (I10,3F10.0)
   22 FORMAT (T25,6I10,1PE20.4)
   23 FORMAT ('1OWN UNIT INFORMATION.',/,'0INITIAL POSITION:',T40,
     1 'COURSE:',T55,'SPEED:',T70,'FREQUENCY COVERAGE',T100,'BINSIZE',
     2 T120,'NR BINS',
     3  /,3F10.1,F16.1,F14.1,1PE20.4,'-',E12.4,E15.4,T120,I7)
   24 FORMAT (' THRESH HOLD =',F8.1,' DBM')
   25 FORMAT ('0TARGET INFORMATION',/'0PLATFORM',T15,'INITIAL POSITION',
     1 T50,'COURSE',T60,'SPEED',//,I5,T15,3F10.1,T50,F6.0,T60,F5.0,/,
     2 '0',T10,'EMITTER',T23,'FREQ',T36,'PRI',T49,'PW',T62,'TRANS POWER'
     3,T75,'ANT GAIN',T88,'BEAM WDTH',T101,'EFF POWER',T114,'ANT SCAN')
   26 FORMAT (I14,T20,1P7E13.4)
   27 FORMAT ('+',T114,'CIR',F5.2,' SPR')
   28 FORMAT (' EMITTER OUT OF RANGE')
   29 FORMAT ('1',T31,'DATA',/,'0',T31,'BEARING',5X,'PULSE',2X,'FREQ BIN
     1',2X,'DBLE PRECISION TIME',4X,'PW',10X,'TOA')
   30 FORMAT (' RCVR ANTENNA GAIN = ',F8.1,' DBM')
   31 FORMAT (' RCVR ANTENNA SCAN = ',F5.2,' SPR')
   32 FORMAT (8(I6,1X))
      END
```

```
      SUBROUTINE UPDATE (I,J)
C
C
C     UPDATE IS CALLED FOR CALCULATION OF ALL TOAS AFTER THE FIRST
C     PULSE FOR EACH EMITTER.  HERE THE EMITTER IS FIRST TESTED FOR
C     JITTER CHARACTERISTICS.  IF JITTER PRESENT COMPUTE RANDOM
C     PULSE INTERVAL IN JITTER PER CENT BOUNDS
C     IF NO JITTER CHARACTERISTICS TEST FOR STAGGER.  IF STAGGER
C     PRESENT COMPUTE PULSE INTERVAL AT PROPER STAGGER RATIO
C     IF NO STAGGER NORMAL CALCULATIONS ARE PERFORMED
C     PLATFORM RANGE AND BEARING ARE UPDATED
C     POWER AND TOA CALCULATIONS ARE PERFORMED
C
C
      COMMON X(10),Y(10),Z(10),VX(10),VY(10),BNG(10),F(10,5),TINT,VXO,VY
     1O,THRESH,PI,PI2,SCNMIN,PRIN,SIGFRQ(10,5),PRI(10,5),PW(10,5),POWER(
     210,5),BW(10,5),SEND(10,5),TNEXT(10,5),SIGNAL(10,5),ANT(10,5),ANG(1
     30,5),XO,YO,ZO,CSEO,SPDO,VL,TWOPI,RDEG,DEGR,TO,TF,FMAX,FMIN,BIN,TT,
     4IF(10,5),PJT(10,5),STGR(10,5),IPT(10,5),ISCAN(10,5),INEXT(10),JNEX
     5T(10),NUNI,NGAUS,NMTR(10),TFINI(10,5),RSPR,RRATE,RBW,RGAIN
      PRIN = PRI(I,J)
      IF (PJT(I,J).EQ.0) GO TO 1
      PRICOM = PRI(I,J)-PJT(I,J)*PRI(I,J)/2.
      CALL RANDU (NUNI,IV,RAND)
      NUNI = IV
      PRIN = PRICOM+RAND*PJT(I,J)*PRI(I,J)
      GO TO 4
    1 IF (STGR(I,J).EQ.0.) GO TO 3
      IF (ISTGR.EQ.1) GO TO 2
      ISTGR = 1
      PRIN = PRI(I,J)-(1.-STGR(I,J))*PRI(I,J)
      GO TO 4
    2 ISTGR = 0
      PRIN = PRI(I,J)-STGR(I,J)*PRI(I,J)
      GO TO 4
    3 PRIN = PRI(I,J)
    4 TFIN = TFINI(I,J)
      TT = SEND(I,J)+PRIN
      IF (TT.LT.TFIN) GO TO 5
      DT = RSPR-RBW*DEGR/RRATE
      TFIN = TFIN+RSPR
      TT = TT+(AINT(DT/PRIN)+1.)*PRIN
    5 TFINI(I,J) = TFIN
      XI = X(I)
      YI = Y(I)
      VXI = VX(I)
      VYI = VY(I)
      XX = XI+VXI*TT
```

```
      YY = YI+VYI*TT
      R = SQRT(XX**2+YY**2)
      POW = POWER(I,J)-20.*ALOG10(R)
      SCNMIN = THRESH-POW-RGAIN
      IF (SCNMIN.GE.0.) GO TO 6
      SIGNAL(I,J) = POW+SCAN(I,J)
      TOF = R/VL
      BNG(I) = ATAN2(XX,YY)
      TNEXT(I,J) = TINT*AINT((TT+TOF)/TINT)
      SEND(I,J) = TT
      RETURN
    6 TNEXT(I,J) = TF+1.
      WRITE (6,7) I,J
      RETURN
C
C
    7 FORMAT (' EMITTER',2I3,'  OUT OF RANGE')
      END
```

```
      FUNCTION SCAN (I,J)
C
C
C     SCAN COMPUTES THE RELATIVE ANGLE OF THE EMITTER ANTENNA AT
C     TRANSMIT TIME.  SIN X /X ATTENUATION IS COMPUTED
C     IF ANTENNA GAIN IS AT OR ABOVE REQUIRED LEVEL THE PROGRAM
C     RETURNS TO CALLING POINT WHERE TOA WILL BE COMPUTED.
C     IF ANTENNA GAIN IS TO LOW TO BREAK RECEIVER THRESHOLD, TIME
C     IS INCREMENTED AT THE PRI RATE UNTIL SUCH TIME AS A
C     SIGNAL BREAKS THRESHOLD
C
C
      COMMON X(10),Y(10),Z(10),VX(10),VY(10),BNG(10),F(10,5),TINT,VXO,VY
     10,THRESH,PI,PI2,SCNMIN,PRIN,SIGFRQ(10,5),PRI(10,5),PW(10,5),POWER(
     210,5),BW(10,5),SEND(10,5),TNEXT(10,5),SIGNAL(10,5),ANT(10,5),ANG(1
     30,5),XO,YO,ZO,CSEO,SPDO,VL,TWOPI,RDEG,DEGR,TO,TF,FMAX,FMIN,BIN,TT,
     4IF(10,5),PJT(10,5),STGR(10,5),IPT(10,5),ISCAN(10,5),INEXT(10),JNEX
     5T(10),NUNI,NGAJS,NMTR(10),TFINI(10,5),RSPR,RRATE,RBW
    1 ANGI = ANG(I,J)
      TFIN = TFINI(I,J)
      BWI = BW(I,J)
      BNGI = BNG(I)
      RATE = ANT(I,J)
      RAT1 = TWOPI/RATE
    2 ANGLE = AMOD((ANGI+TT*RAT1-PI-BNGI),TWOPI)
      IF (ANGLE.GT.PI) ANGLE=ANGLE-TWOPI
      IF (ANGLE.LT.(-PI)) ANGLE=TWOPI+ANGLE
      IF (ABS(ANGLE).LE.1.E-2) GO TO 5
      U = BWI*ANGLE
      SCAN = 10.*ALOG10((SIN(U)/U)**2)
      IF (SCAN.GE.SCNMIN) GO TO 4
      TT = TT+PRIN
      IF (TT.GE.TFIN) GO TO 3
      GO TO 2
    3 DT = TFIN+RSPR-RBW*DEGR/RRATE
      TFIN = TFIN+RSPR
      TFINI(I,J) = TFIN
      TT = TT+(AINT(DT/PRIN)+1.)*PRIN
      IF (TT.GE.TF) GO TO 4
      GO TO 2
    4 RETURN
    5 SCAN = 0.0
      RETURN
      END
```

```
OWN UNIT INFORMATION.

INITIAL POSITION:                     COURSE:        SPEED:          FREQUENCY COVERAGE              BINSIZE          NR BINS
     0.0         0.0        0.0         90.0          250.0           8.0000E 09-  1.2000E 10     3.1250E 07           128
THRESH HOLD =    -70.0 DBM
RCVR ANTENNA GAIN =      25.0 DBM
RCVR ANTENNA SCAN =   2.00 SPR

TARGET INFORMATION

PLATFORM     INITIAL POSITION                    COURSE     SPEED

   1            -98.0       23.0       0.0         305.       15.

       EMITTER       FREQ         PRI           PW            TRANS POWER  ANT GAIN     BEAM WDTH    EFF POWER    ANT SCAN
          1       9.9950E 09   4.9000E-03    1.0000E-06    1.5000E 05   2.5000E 01   1.0000E 00   5.5099E 01   CIR 1.00 SPR
          2       8.8000E 09   4.6000E-03    1.5000E-06    2.0000E 05   2.0000E 01   2.0000E 00   5.2454E 01   CIR 6.00 SPR
          3       9.0000E 09   2.7000E-03    2.5000E-06    5.0000E 04   2.3000E 01   5.0000E 00   4.9238E 01   CIR10.00 SPR
TARGET INFORMATION

PLATFORM     INITIAL POSITION                    COURSE     SPEED

   2           -100.0       24.0       0.0          0.         0.

       EMITTER       FREQ         PRI           PW            TRANS POWER  ANT GAIN     BEAM WDTH    EFF POWER    ANT SCAN
          1       9.9950E 09   3.0000E-03    2.0000E-06    5.0000E 05   2.0000E 01   1.0000E 01   5.5328E 01   CIR 2.00 SPR
          2       8.5000E 09   4.5000E-03    2.2000E-06    1.5000E 05   1.5000E 01   1.5000E 01   4.6506E 01   CIR10.00 SPR
          3       9.5000E 09   4.5000E-03    1.8000E-06    2.5000E 05   1.5000E 01   5.0000E 00   4.7758E 01   CIR 5.00 SPR
TARGET INFORMATION

PLATFORM     INITIAL POSITION                    COURSE     SPEED

   3           -100.0       25.0       0.0          35.       15.

       EMITTER       FREQ         PRI           PW            TRANS POWER  ANT GAIN     BEAM WDTH    EFF POWER    ANT SCAN
          1       8.8850E 09   9.5000E-04    4.1000E-06    1.0000E 05   2.2000E 01   7.5000E 00   5.1360E 01   CIR 1.00 SPR
          2       1.0147E 10   7.6660E-04    7.5000E-07    5.0000E 04   3.5000E 01   1.5000E 00   6.0196E 01   CIR 0.25 SPR
          3       8.7000E 09   1.2500E-03    5.0000E-06    5.0000E 05   2.0000E 01   2.5000E 01   5.6533E 01   CIR 1.00 SPR
TARGET INFORMATION

PLATFORM     INITIAL POSITION                    COURSE     SPEED

   4           -100.0       26.0       0.0          45.       12.

       EMITTER       FREQ         PRI           PW            TRANS POWER  ANT GAIN     BEAM WDTH    EFF POWER    ANT SCAN
          1       9.1880E 09   5.0453E-03    2.3560E-06    5.0000E 06   1.5000E 01   5.1000E 00   6.1059E 01   CIR 0.10 SPR
          2       9.2500E 09   1.1000E-03    1.0000E-06    5.0000E 04   3.0000E 01   5.0000E-01   5.6000E 01   CIR 0.10 SPR
          3       8.6000E 09   1.5000E-03    1.5000E-06    5.0000E 04   2.5000E 01   6.0000E 00   5.1633E 01   CIR 8.00 SPR
          4       8.9000E 09   2.0000E-03    1.4000E-06    3.0000E 05   1.0000E 01   5.0000E 00   4.4117E 01   CIR 4.00 SPR
          5       9.6000E 09   1.2000E-03    1.6000E-06    3.5000E 05   1.5000E 01   4.0000E 00   4.9129E 01   CIR 2.00 SPR
TARGET INFORMATION

PLATFORM     INITIAL POSITION                    COURSE     SPEED

   5           -200.0       50.0   50000.0          60.       14.

       EMITTER       FREQ         PRI           PW            TRANS POWER  ANT GAIN     BEAM WDTH    EFF POWER    ANT SCAN
          1       9.6400E 09   5.0000E-04    5.0000E-07    2.5000E 05   1.5000E 01   1.0000E 01   4.7631E 01   CIR 2.00 SPR
          2       9.9000E 09   2.0000E-03    5.0000E-06    3.0000E 05   1.5000E 01   1.5000E 01   4.8192E 01   CIR 5.00 SPR
TARGET INFORMATION

PLATFORM     INITIAL POSITION                    COURSE     SPEED

   6           -200.0       48.0       0.0         240.       20.

       EMITTER       FREQ         PRI           PW            TRANS POWER  ANT GAIN     BEAM WDTH    EFF POWER    ANT SCAN
          1       9.2750E 09   1.1000E-03    1.0000E-06    6.5000E 04   2.4000E 01   4.6000E 00   5.1116E 01   CIR 0.36 SPR
          2       9.5000E 09   4.0000E-03    1.0000E-06    5.0000E 04   3.0000E 01   5.0000E-01   5.5769E 01   CIR10.00 SPR
```

DATA

BEARING	PULSE	FREQ BIN	DBLE PRECISION	TIME	PW	TOA
280	1	32	0	7381	25	1.8896E-01
281	2	41	0	7421	10	1.8999E-01
281	3	16	0	7446	22	1.9064E-01
281	4	64	0	7453	10	1.9080E-01
281	5	48	0	7454	10	1.9083E-01
281	6	41	0	7464	10	1.9109E-01
281	7	26	0	7470	15	1.9125E-01
281	8	16	0	7482	22	1.9154E-01
281	9	64	0	7483	20	1.9158E-01
281	10	32	0	7486	25	1.9166E-01
281	11	48	0	7503	18	1.9208E-01
281	12	41	0	7507	10	1.9219E-01
281	13	53	0	7537	5	1.9296E-01
281	14	29	0	7547	41	1.9322E-01
281	15	41	0	7550	10	1.9329E-01
281	16	53	0	7556	5	1.9346E-01
281	17	29	0	7558	41	1.9350E-01
281	18	69	0	7558	7	1.9351E-01
281	19	53	0	7576	5	1.9396E-01
281	20	23	0	7579	50	1.9405E-01
281	21	29	0	7584	41	1.9417E-01
281	22	69	0	7588	7	1.9427E-01
281	23	32	0	7592	25	1.9436E-01
281	24	41	0	7593	10	1.9439E-01
281	25	29	0	7595	41	1.9445E-01
281	26	53	0	7596	5	1.9446E-01
281	27	64	0	7600	20	1.9458E-01
281	28	61	0	7604	50	1.9468E-01
281	29	48	0	7610	10	1.9483E-01
281	30	53	0	7615	5	1.9496E-01
281	31	69	0	7618	7	1.9504E-01
281	32	29	0	7621	41	1.9512E-01
281	33	16	0	7622	22	1.9514E-01
281	34	23	0	7628	50	1.9530E-01
281	35	29	0	7632	41	1.9540E-01
281	36	53	0	7635	5	1.9546E-01
282	37	41	0	7636	10	1.9549E-01
282	38	29	0	7644	41	1.9569E-01
282	39	64	0	7644	10	1.9570E-01
282	40	69	0	7648	7	1.9581E-01
282	41	53	0	7654	5	1.9596E-01
282	42	40	0	7662	10	1.9616E-01
282	43	26	0	7663	15	1.9619E-01
282	44	29	0	7670	41	1.9635E-01
282	45	53	0	7674	5	1.9646E-01

APPENDIX C

PULSE TRAIN SEPARATOR SIMULATION PROGRAM PL/ATAC LISTING

Program variables are described as they are declared in the program.

SIMPLE PL/ATAC COMPILER VERSION OF FEBRUARY 1, 1974.   CLOCK TIME = 9:51:33.79.

TODAY IS MARCH 18, 1975.   CLOCK TIME = 22:57:29.51.

```
/"****
       TOGGLES: $PUNCH
****"/

/"* PL/ATAC PULSE TRAIN SEPARATOR



       *****  N O T E  *****

       THIS VERSION OF PL/ATAC CONTAINS NO DOUBLE PRECISION CAPABILITY
       NECESSITATING EMPLOYMENT OF THE 'CODE' COLLECTION HEAD WHERE
       DOUBLE PRECISION CALCULATIONS ARE PERFORMED.

       **********************

       PROGRAM IS DESIGNED TO SIMULATE A HARDWARE SIGNAL PREPROCESSING
       SYSTEM WITH A CAPABILITY OF SEPARATING UP TO A MAXIMUM OF TEN
       STABLE PULSE TRAINS FROM THE OUTPUT OF AN IFM RECEIVER.  THE IFM
       RECEIVER OPERATES IN CONJUNCTION WITH A HIGH GAIN SCANNING
       ANTENNA.  FOR SIMULATION PURPOSES, DIRECTION OF ARRIVAL (DOA) IS
       ASSUMED TO BE OBTAINED FROM AN AZIMUTH RESOLVER GATED BY THE
       PULSE TRAIN SEPERATOR(PTS) HARDWARE.  THE OUTPUT PULSES FROM THE
       IFM RECEIVER FED TO THE PTS ARE SIMULATED BY LOADING A SERIES
       OF FIVE WORDS FOR EACH PULSE RECEIVED.
            1.  DOA - IN DEGREES  LSB = 1 DEG
            2.  FREQUENCY - IN TERMS OF FREQUENCY BIN OF THE PULSE
          3,4.  TIME OF ARRIVAL(TOA) - DOUBLE PRECISION VALUE
                                   LSB = 25.6 MICROSECONDS.
            5.  PULSE WIDTH(PW) - LSB =.1 MICROSECOND
       UPON RECEIPT OF FIVE PULSES OF A STABLE TRAIN, THE PTS LOCKS
       ON THE TRAIN AND CONTINUES TO TRACK THE TRAIN FOR ONE ANTENNA
       BEAMWIDTH.  THE PULSE REPETITION INTERVAL(PRI) IS COMPUTED AND
       AZIMUTH CENTROIDING, BASED ON DOA AND TOA OF THE FIRST AND LAST
       PULSE OF THE TRAIN, IS EMPLOYED TO COMPUTE THE BEARING OF THE
       EMITTER ASSOCIATED WITH THE TRAIN.  THE PARAMETERS ASSOCIATED
       WITH THE TRAIN, OUTPUT TO A MAINFRAME COMPUTER FOR SIGNAL
       CLASSIFICATION, CONSIST OF A SERIES OF SIX WORDS.
            1.  EMITTER BEARING
            2.  FREQUENCY
          3,4.  TIME FIRST SEEN
            5.  PULSE WIDTH
            6.  PULSE REPETITION INTERVAL
       THIS OUTPUT IS SIMULATED BY STORING THE PARAMETERS IN A MEMORY
       BUFFER.
       RESIDUAL PULSES PASS THROUGH A SEPARATE CHANNEL OF THE PTS TO A
       HIGH SPEED MINI COMPUTER WHERE THEY ARE PROCESSED TO DETERMINE
       PRI, JITTER AND STAGGER CHARACTERISTICS IF THEY EXIST.  THIS
       ACTION IS SIMULATED BY COMPRESSING THE RESIDUAL DATA WITHIN THE
       INPUT BUFFER TO BE PROCESSED BY AN ATAC ASSEMBLY LANGUAGE
       PROGRAM WHICH MAY BE INCORPORATED USING THE CODE COLLECTION HEAD
"/


/"****  G L O B A L   V A R I A B L E S   ****"/

/"  DESCRIPTION OF GLOBAL VARIABLES

            RAM VARIABLES

       BUF    4000 WORD INPUT BUFFER
       NS     NEW SIGNAL POINTER FILE.  HOLDS POINTERS TO A MAXIMUM OF 5
              PULSES FOR EACH EMITTER UP TO 40 EMITTERS.
       TMIN   DOUBLE PRECISION TIME FIRST SEEN
       NC     PTS ACTIVE CHANNEL COUNTER.  INITIAL VALUE DETERMINES
              NUMBER OF CHANNELS OF PTS.  MAX NO CHANNELS = 10-NC INIT
       PI     TIME INTERVAL BETWEEN PULSES
       PRI    PULSE REPETITION INTERVAL
       CHAN   240 WORD BUFFER IN WHICH STABLE PULSE TRAIN PARAMETERS ARE
              STORED
       CNDX   INDEX TO CHAN
       CINDX  INDEX TO CHAN
       DP     DATA POINTER
       SSF    STABLE SIGNAL FOUND
       NSSF   NON-STABLE SIGNAL FOUND


```

```
(
(
(          REGISTER VARIABLES
(
(     I,J    NS FILE INDEXES
(     BP     BUFFER POINTER
("/
(
(DECLARE BUF(3999)RAM
(DECLARE NS(199)RAM
(DECLARE TMIN(79)RAM
(DECLARE(NC,PI(5),PRI(1),CNDX)RAM
(DECLARE CHAN(120)RAM
(DECLARE NSSF(39)RAM
(DECLARE SSF(39) RAM
(DECLARE CINDX(39) RAM
(DECLARE (BP,I,J)REGISTER
(DECLARE DP RAM
(
(DEFINE TRUETBL=(FALSE,TRUE)
(
(
(LOCAL PROCEDURE PRICOMP
(
(/"  PRICOMP DETERMINES THE STABILITY OF A PULSE TRAIN.  IF STABLE
(    THE PRI IS COMPUTED AND THE TRAIN IS FLAGGED AS A STABLE TRAIN.
(    ALL PARAMETERS EXCEPT DF ARE LOADED INTO THE CHANNEL BUFFER.
(    IF THE TRAIN IS NON STABLE IT IS FLAGGED AS SUCH.
(
(          DESCRIPTION OF VARIABLES
(
(    K,KK    INDEXES FOR CALCULATIONS
(    MP      MISSING PULSE COUNTER
(    PRITEM CALCULATION VALUE OF PRI
("/
(
( DECLARE (K,MP,KK,KJ,PRITEM)RAM
(   K=2
(   KK=KJ=1
(   MP=0
(
(/"  TEST PULSE INTERVALS TO DETERMINE STABILITY OF TRAIN. A PULSE
(    TRAIN IS CONSIDERED STABLE IF NO MORE THAN TWO MISSING PULSES
(    ARE ENCOUNTERED IN INITIAL PULSES AND VARIATIONS IN TIME
(    INTERVAL BETWEEN PULSES DOES NOT EXCEED +- 51.2 MICROSECONDS.  "/
(
(
(LOOP1:
(%   DO
(   TEST@MP>2
(      IFSO: LEAVE LOOP1
(   END
(   TEST @PI(.(@K-1).)/@KK IN (@PI(.@K.)/@KJ-2) TO (@PI(.@K.)/@KJ+2)
(      ABOVE: DO
(             MP=@MP+1
(             KK=@KK+1
(             REPEAT LOOP1
(             END
(      BELOW: DO
(             MP=@MP+1
(             KJ=@KJ+1
(             REPEAT LOOP1
(             END
(      END
(   K=@K+1
(   KK=@KJ
(   KJ=1
(   TEST @K>4
(      IFNOT: REPEAT LOOP1
(   END
(
(/"  COMPUTE PRI AND TEST FOR ROUND OFF ERROR.  IF ERROR EXISTS
(    ROUND UP BY 1.
("/
(
(   K=1
(   PRI=0
(   ITERATE 4 TIMES
(   PRI=@PRI+@PI(.@K.)
(   K=@K+1
(   END
(   PRITEM=@PRI
(   PRI=@PRI/(4+@MP)
(   TEST @PRITEM MOD(4+@MP)>(4+@MP)/2
(      IFSO: PRI=@PRI+1
(   END
(
(/"  LOAD PARAMETERS INTO CHANNEL BUFFER  "/
(
(   DP=@NS(.@I.)
(   CNDX=@CNDX+6
(   CINDX(.@I/5.)=@CNDX
(   CHAN(.(@CNDX+1).)=@BUF(.(@DP+1).)
(   CHAN(.(@CNDX+2).)=@TMIN(.(@I/5*2).)
(   CHAN(.(@CNDX+3).)=@TMIN(.(@I/5*2+1).)
(   CHAN(.(@CNDX+4).)=@BUF(.(@DP+4).)
(   CHAN(.(@CNDX+5).)=@PRI
(
(/"  FLAG PULSE TRAIN AS STABLE AND INCREMENT CHANNEL COUNTER  "/
(
(   SSF(.(@I/5).)=TRUE
(   NC=@NC+1
(   RETURN
(END LOOP1
(
```

```
/"  TRAIN FAILED STABILITY TEST. FLAG TRAIN AS NON STABLE  "/

   NSSF(.(@I/5).)=TRUE
END PRICOMP


PROCEDURE MAIN

/"  DESCRIPTION OF VARIABLES

    BRNGF   RCVR ANTENNA BEARING AT TOA OF FIRST PULSE OF A SIGNAL
    BRNGL   RCVR ANTENNA BEARING AT TOA OF LAST PULSE OF A TRAIN
    CON1    DOUBLE PRECISION TIME CALCULATION VARIABLE
    CON2    DOUBLE PRECISION TIME CALCULATION VARIABLE
    II,JJ   INDICES
    SCANTM  RCVR ANTENNA SCAN TIME FOR ONE BEAMWIDTH
    TCAL    LOWER HALF DOUBLE PRECISION TIME OF ARRIVAL
    TCAU    UPPER HALF DOUBLE PRECISION TIME OF ARRIVAL
    TMAX    EXPECTED TIME OF LAST PULSE INTERCEPT
    TMIN    DOUBLE PRECISION TIME OF ARRIVAL OF FIRST PULSE
    TTEM    DOUBLE PRECISION TIME CALCULATION VARIABLE
    TTEST   DOUBLE PRECISION TIME CALCULATION VARIABLE
"/

DECLARE(BRNGF,BRNGL(39),SCANTM(1),II,JJ,STR)RAM
DECLARE(TCAL,TCAU,TMAX(79),TTEST(1),TTEM(1))RAM
DECLARE(CON1(1),CON2(1))RAM

/"  INITIALIZE VARIABLES  "/

   CNDX=-6
   SCANTM=1120
   I=0
   BP=0
   NC=1

/"  SET NEW SIGNAL POINTER FILE TO NEGATIVE VALUES  "/

   ITERATE 200 TIMES
   NS(.@I.)=-1
   I=@I+1
   END
   I=0
   NS(.@I.)=0

/"  LOAD INPUT BUFFER FROM CARDS  "/

   READ_DEC(BUF,BUF+3999)

MAINLP:
%  DO

/"  LOAD DP TOA OF FIRST PULSE OF SIGNAL FROM INPUT BUFFER AND
    COMPUTE EXPECTED TIME OF ARRIVAL OF LAST PULSE BASED ON RCVR
    ANTENNA SCAN TIME FOR ONE ANTENNA BEAMWIDTH  "/

   TCAL=@BUF(.(@BP+2).)
   TCAU=@BUF(.(@BP+3).)
   CODE
         LDRM  D,14,TCAL,2
         DADD  D,14,SCANTM
         STRM  D,14,TTEM,2
   END
   STR=@I/5*2
   TMAX(.@STR.)=@TTEM
   TMAX(.(@STR+1).)=@TTEM(.1.)
   TMIN(.@STR.)=@TCAL
   TMIN(.(@STR+1).)=@TCAU
COMPUTE:
%  DO

/"  INCREMENT BUFFER POINTER AND CHECK AT BUFFER LIMIT.  IFSO, GO
    COMPRESS RESIDUAL PULSES WITHIN INPUT BUFFER FOR FUTHER PROCESSING
"/

   BP=@BP+5
   TEST @BP>3995
      IFSO: LEAVE MAINLP
   END

/"  IF NO SIGNAL IN BUFFER INDEXED BY BUFFER POINTER INCREMENT
    POINTER AND REPEAT COMPUTE  "/

   TEST @BUF(.@BP.)
      ZERO: REPEAT COMPUTE
   END
   I=0
   J=0
SORT:
%  DO

/"  IF AT END OF NEW SIGNAL POINTER FILE, LEAVE SORT  "/

   TEST @I>195
      IFSO: LEAVE SORT
   END

/"  IF NEW SIGNAL POINTER FILE INDEXED BY I IS EMPTY INDEX AND
    CHECK NEXT LOCATION  "/

   TEST @NS(.@I.)
      NEG: DO
         I=@I+5
         REPEAT SORT
```

```
        END
     END

/"   COMPARE TOA OF BUFFER SIGNAL WITH TMAX OF SIGNAL FOUND IN NEW
     SIGNAL POINTER FILE.  IF TOA OF BUFFER SIGNAL IS GREATER THAN
     OR EQUAL TO TMAX, TEST FOR STABLE SIGNAL IN NS POINTER FILE.
     IF  SIGNAL HAS BEEN CLASSIFIED STABLE, RESET ACTIVE CHANNEL
     COUNTER AND SSF FLAG, ZERO FIRST FIVE PULSES FROM THIS SIGNAL
     IN INPUT BUFFER AND COMPUTE EMITTER BEARING USING FIRST AND
     DOA OF SIGNAL "/

     CON1=@BUF(.(@BP+2).)
     CON1(.1.)=@BUF(.(@BP+3).)
     CON2=@TMAX(.(@I/5*2).)
     CON2(.1.)=@TMAX(.(@I/5*2+1).)
     CODE
            LDRM   D,14,CON1,2
            LDRM   D,7,CON2,2
            DSUB   R,7,14
            STRM   D,7,TTEST,2
     END
     TEST @TTEST
        NEG,ZERO: DO
            TEST @SSF(.(@I/5).)
                 POS: DO
                       NC=@NC-1
                       SSF(.(@I/5).)=FALSE
                       BRNGF=@BUF(.(@NS(.@I.)).)
                       II=0
                      ITERATE 5 TIMES
                        JJ=0
                        ITERATE 5 TIMES
                          BUF(.(@NS(.(@I+@II).)+@JJ).)=0
                          JJ=@JJ+1
                        END
                        II=@II+1
                      END
                       CHAN(.@CINDX(.@I/5.).)=(@BRNGF+@BRNGL(.@I/5.))/2
                       END
            END

/"   IF NS POINTER FILE SIGNAL HAS BEEN CLASSIFIED AS NON STABLE
     RESET NSSF FLAG  "/

            NSSF(.(@I/5).)=FALSE

/"   CLEAR THE SIGNAL FROM NS POINTER FILE  "/

           II=0
           ITERATE 5 TIMES
           NS(.(@I+@II).)=-1
           II=@II+1
           END
           I=@I+5
           REPEAT SORT
        END
     END
     TEST @BUF(.(@BP+1).)==@BUF(.(@NS(.@I.)+1).)
        IFNOT: DO
            I=@I+5
            REPEAT SORT
            END
     END

/"   TEST FOR MATCH BETWEEN BUFFER SIGNAL FREQ AND PW AND NS SIGNAL
     POINTER FILE FREQ AND PW.  IF BOTH DO NOT MATCH REPEAT SORT.
     IF MATCH OCCURS AND SIGNAL HAS BEEN CLASSIFIED AS STABLE,
     ZERO BUFFER SIGNAL AND REPEAT COMPUTE.  IF SIGNAL HAS BEEN
     CLASSIFIED AS NON STABLE, REPEAT COMPUTE.  "/

     TEST @BUF(.(@BP+4).)==@BUF(.(@NS(.@I.)+4).)
        IFNOT: DO
            I=@I+5
            REPEAT SORT
            END
     END
     TEST @SSF(.(@I/5).)
        NONZERO: DO
            BRNGL(.@I/5.)=@BUF(.@BP.)
            II=0
            ITERATE 5 TIMES
            BUF(.(@BP+@II).)=0
            II=@II+1
            END
        REPEAT COMPUTE
        END
     END
     TEST @NSSF(.(@I/5).)==TRUE
        IFSO: REPEAT COMPUTE
     END

/"   IF SIGNAL HAS NOT BEEN CLASSIFIED, STORE BUFFER POINTER IN
     NS POINTER FILE. IF THIS IS THE FIFTH PULSE ASSOCIATED WITH
     THIS TRAIN, COMPUTE TIME INTERVALS BETWEEN PULSES AND CALL
     FOR PRI COMPUTATION.   "/

ITERATE 3 TIMES
   J=@J+1
   TEST @NS(.(@I+@J).)
      NEG: DO
            NS(.(@I+@J).)=@BP
            REPEAT COMPUTE
            END
     END
     END
```

```
    NS(.(@I+4).)=@BP
    TEST @NC==10
       IFSO: DO
          NSSF(.(@I/5).)=TRUE
          REPEAT COMPUTE
          END
    END
    J=1
    ITERATE 4 TIMES
    PI(.@J.)=@BUF(.(@NS(.(@I+@J).)+2).)-@BUF(.(@NS(.(@I+@J-1).)+2).)
    TEST @PI(.@J.)
       NEG:    PI(.@J.)=@PI(.@J.)+32767
    END
    J=@J+1
    END
    CALL PRICOMP()
    REPEAT COMPUTE
END SORT
    I=0
    CYCLE
    TEST @NS(.@I.)
       NEG: DO
          NS(.@I.)=@BP
          REPEAT MAINLP
          END
    END
    TEST(I=@I+5)>195
       IFSO: DO
             DUMP_DEC(NS,NS+195)
             RETURN
             END
    END
    END
END COMPUTE
END MAINLP

/"  ROUTINE BELOW COMPRESSES RESIDUAL PULSES WITHIN THE INPUT
    BUFFER. AN ATAC ASSEMBLY LANGUAGE PROGRAM TO PROCESS THIS
    RESIDUAL MAY BE INCORPORATED BETWEEN THE END AND END MAIN
    STATEMENTS UTILIZING THE CODE COLLECTION HEAD.  "/

    BP=0
    ZEROCK:
%      CYCLE
    TEST @BUF(.@BP+1.)
       ZERO:  DO
          DP=@BP
          LEAVE ZEROCK
       END
    END
    BP=@BP+5
END ZEROCK
    CYCLE
       BP=@BP+5
       TEST @BP>3995
          IFSO: DO
             RETURN
             END
       END
          TEST @BUF(.(@BP+1).)==0
             IFNOT: DO
                  I=0
               ITERATE 5 TIMES
                  BUF(.(@DP+@I).)=@BUF(.(@BP+@I).)
                  BUF(.(@BP+@I).)=0
                  I=@I+1
                 END
                 DP=@DP+5
                 END
          END
    END
END MAIN


/"****  LINK MODULE FOR EXECUTING PL/ATAC PROGRAM.
       $EXECUTIVE
****"/
   MODULE LINK
      DECLARE PROCSTK(64)RAM
      RO = PROCSTK
      CALL MAIN()
   END LINK
EOF EOF EOF EOF EOF EOF EOF EOF EOF
```

APPENDIX D

PULSE TRAIN SEPARATOR SIMULATION PROGRAM ASSEMBLY LANGUAGE LISTING AND OUTPUT

```
LOC   OBJECT CODE     CARD IMAGE                                                        CARDNUM
                      .****"/
                      .
                      ./"* PL/ATAC PULSE TRAIN SEPARATOR
                      .
                      .
                      .     *****  N O T E  *****
                      .
                      .    THIS VERSION OF PL/ATAC CONTAINS NO DOUBLE PRECISION CAPABILITY
                      .    NECESSITATING EMPLOYMENT OF THE 'CODE' COLLECTION HEAD WHERE
                      .    DOUBLE PRECISION CALCULATIONS ARE PERFORMED.
                      .
                      .    ********************
                      .
                      .    PROGRAM IS DESIGNED TO SIMULATE A HARDWARE SIGNAL PREPROCESSING
                      .    SYSTEM WITH A CAPABILITY OF SEPARATING UP TO A MAXIMUM OF TEN
                      .    STABLE PULSE TRAINS FROM THE OUTPUT OF AN IFM RECEIVER.  THE IFM
                      .    RECEIVER OPERATES IN CONJUNCTION WITH A HIGH GAIN SCANNING
                      .    ANTENNA.  FOR SIMULATION PURPOSES, DIRECTION OF ARRIVAL (DOA) IS
                      .    ASSUMED TO BE OBTAINED FROM AN AZIMUTH RESOLVER GATED BY THE
                      .    PULSE TRAIN SEPERATOR(PTS) HARDWARE.  THE OUTPUT PULSES FROM THE
                      .    IFM RECEIVER FED TO THE PTS ARE SIMULATED BY LOADING A SERIES
                      .    OF FIVE WORDS FOR EACH PULSE RECEIVED.
                      .         1.  DOA - IN DEGREES  LSB = 1 DEG
                      .         2.  FREQUENCY - IN TERMS OF FREQUENCY BIN OF THE PULSE
                      .       3,4.  TIME OF ARRIVAL(TOA) - DOUBLE PRECISION VALUE
                      .                                   LSB = 25.6 MICROSECONDS.
                      .         5.  PULSE WIDTH(PW) - LSB =. 1 MICROSECOND
                      .    UPON RECEIPT OF FIVE PULSES OF A STABLE TRAIN, THE PTS LOCKS
                      .    ON THE TRAIN AND CONTINUES TO TRACK THE TRAIN FOR ONE ANTENNA
                      .    BEAMWIDTH.  THE PULSE REPETITION INTERVAL(PRI) IS COMPUTED AND
                      .    AZIMUTH CENTROIDING, BASED ON DOA AND TOA OF THE FIRST AND LAST
                      .    PULSE OF THE TRAIN, IS EMPLOYED TO COMPUTE THE BEARING OF THE
                      .    EMITTER ASSOCIATED WITH THE TRAIN.  THE PARAMETERS ASSOCIATED
                      .    WITH THE TRAIN, OUTPUT TO A MAINFRAME COMPUTER FOR SIGNAL
                      .    CLASSIFICATION, CONSIST OF A SERIES OF SIX WORDS.
                      .         1.  EMITTER BEARING
                      .         2.  FREQUENCY
                      .       3,4.  TIME FIRST SEEN
                      .         5.  PULSE WIDTH
                      .         6.  PULSE REPETITION INTERVAL
                      .    THIS OUTPUT IS SIMULATED BY STORING THE PARAMETERS IN A MEMORY
                      .    BUFFER.
                      .    RESIDUAL PULSES PASS THROUGH A SEPARATE CHANNEL OF THE PTS TO A
                      .    HIGH SPEED MINI COMPUTER WHERE THEY ARE PROCESSED TO DETERMINE
                      .    PRI, JITTER AND STAGGER CHARACTERISTICS IF THEY EXIST.  THIS
                      .    ACTION IS SIMULATED BY COMPRESSING THE RESIDUAL DATA WITHIN THE
                      .    INPUT BUFFER TO BE PROCESSED BY AN ATAC ASSEMBLY LANGUAGE
                      .    PROGRAM WHICH MAY BE INCORPORATED USING THE CODE COLLECTION HEAD
                      ."/
                      .
                      .
                      ./"****  G L O B A L   V A R I A B L E S  ****"/
                      .
                      ./"  DESCRIPTION OF GLOBAL VARIABLES
```

```
LOC   OBJECT CODE      CARD IMAGE
                       .
                       .          RAM VARIABLES
                       .
                       .     BUF    4000 WORD INPUT BUFFER
                       .     NS     NEW SIGNAL POINTER FILE.  HOLDS POINTERS TO A MAXIMUM OF 5
                       .            PULSES FOR EACH EMITTER UP TO 40 EMITTERS.
                       .     TMIN   DOUBLE PRECISION TIME FIRST SEEN
                       .     NC     PTS ACTIVE CHANNEL COUNTER.  INITIAL VALUE DETERMINES
                       .            NUMBER OF CHANNELS OF PTS.  MAX NO CHANNELS = 10-NC INIT
                       .     PI     TIME INTERVAL BETWEEN PULSES
                       .     PRI    PULSE REPETITION INTERVAL
                       .     CHAN   240 WORD BUFFER IN WHICH STABLE PULSE TRAIN PARAMETERS ARE
                       .            STORED
                       .     CNDX   INDEX TO CHAN
                       .     CINDX  INDEX TO CHAN
                       .     DP     DATA POINTER
                       .     SSF    STABLE SIGNAL FOUND
                       .     NSSF   NON-STABLE SIGNAL FOUND
                       .
                       .
                       .
                       .
                       .
                       .          REGISTER VARIABLES
                       .
                       .     I,J    NS FILE INDEXES
                       .     BP     BUFFER POINTER
                       ."/
                       .
                       .DECLARE BUF(3999)RAM
                                ORG   RAM
2000                   BUF      DS    4000
                       .DECLARE NS(199)RAM
2FA0                   NS       DS    200
                       .DECLARE TMIN(79)RAM
3068                   TMIN     DS    80
                       .DECLARE(NC,PI(5),PRI(1),CNDX)RAM
3088                   NC       DS    1
3089                   PI       DS    6
308F                   PRI      DS    2
30C1                   CNDX     DS    1
                       .DECLARE CHAN(120)RAM
30C2                   CHAN     DS    121
                       .DECLARE NSSF(39)RAM
313B                   NSSF     DS    40
                       .DECLARE SSF(39) RAM
3163                   SSF      DS    40
                       .DECLARE CINDX(39) RAM
318B                   CINDX    DS    40
                       .DECLARE (BP,I,J)REGISTER
                       .DECLARE DP RAM
31B3                   DP       DS    1
                       .
```

```
LOC   OBJECT CODE      CARD IMAGE
                       .DEFINE TRUETBL=(FALSE,TRUE)
                       FALSE     EQU    0
                       TRUE      EQU    1
                       TRUETBL   EQU    2
                       .
                       .
                       .LOCAL PROCEDURE PRICOMP
                       .
                       ./"  PRICOMP DETERMINES THE STABILITY OF A PULSE TRAIN.  IF STABLE
                       .    THE PRI IS COMPUTED AND THE TRAIN IS FLAGGED AS A STABLE TRAIN.
                       .    ALL PARAMETERS EXCEPT DF ARE LOADED INTO THE CHANNEL BUFFER.
                       .    IF THE TRAIN IS NON STABLE IT IS FLAGGED AS SUCH.
                       .
                       .         DESCRIPTION OF VARIABLES
                       .
                       .    K,KK   INDEXES FOR CALCULATIONS
                       .    MP     MISSING PULSE COUNTER
                       .    PRITEM CALCULATION VALUE OF PRI
                       ."/
                       .
                       . DECLARE (K,MP,KK,KJ,PRITEM)RAM
31B4                   K         DS     1
31B5                   MP        DS     1
31B6                   KK        DS     1
31B7                   KJ        DS     1
31B8                   PRITEM    DS     1
                       .   K=2
                                 ORG    ROM
                       PRICOMP   EQU    $
0000  400C                       LDR    IS,12,0
0001  4022                       LDR    IS,2,2
0002  9C02 31B4                  STR    D,2,0+K
                       .   KK=KJ=1
0004  4012                       LDR    IS,2,1
0005  9C02 31B7                  STR    D,2,0+KJ
0007  9C02 31B6                  STR    D,2,0+KK
                       .   MP=0
0009  9C0C 31B5                  STR    D,12,0+MP
                       .
                       ./"  TEST PULSE INTERVALS TO DETERMINE STABILITY OF TRAIN. A PULSE
                       .    TRAIN IS CONSIDERED STABLE IF NO MORE THAN TWO MISSING PULSES
                       .    ARE ENCOUNTERED IN INITIAL PULSES AND VARIATIONS IN TIME
                       .    INTERVAL BETWEEN PULSES DOES NOT EXCEED +- 51.2 MICROSECONDS.  "/
                       .
                       .
                       .LOOP1:
                       .%   DO
                       @0        EQU    $
                       .   TEST@MP>2
000B  E202 31B5                  LDR    D,2,0+MP
000D  2022                       CMP    IS,2,2
                       .      IFSO: LEAVE LOOP1
                       @4        EQU    $
```

```
LOC    OBJECT CODE      CARD IMAGE                                                    CARDNUM

000E   C106 0012                 BRC    I,6,@5                                           161
0010   C107 00F1                 BRC     I,7,@1                                          162
                    .       END                                                          163
                    @5           ECU    $                                                164
                    @2           EQU    $                                                165
                    .       TEST @PI(.(@K-1).)/@KK IN (@PI(.@K.)/@KJ-2) TO (@PI(.@K.)/@KJ+2)   166
0012   E202 31B4                 LDR    D,2,0+K                                          167
0014   6FF2                      ADD    IS,2,-1                                          168
0015   E324 30B9                 LDR    DX,4,0+PI,2                                      169
0017   D8F3                      SHD    RA,3,16                                          170
0018   E205 31B6                 LDR    D,5,0+KK                                         171
001A   FC53                      DIV    R,3,5                                            172
001B   E205 31B4                 LDR    D,5,0+K                                          173
001D   E357 30B9                 LDR    DX,7,0+PI,5                                      174
001F   D8F6                      SHD    RA,6,16                                          175
0020   E208 31B7                 LDR    D,8,0+KJ                                         176
0022   FC86                      DIV    R,6,8                                            177
0023   6FE7                      ADD    IS,7,-2                                          178
0024   E208 31B4                 LDR    D,8,0+K                                          179
0026   E383 30B9                 LDR    DX,3,0+PI,8                                      180
0028   D8F2                      SHD    RA,2,16                                          181
0029   E205 31B7                 LDR    D,5,0+KJ                                         182
002B   FC52                      DIV    R,2,5                                            183
002C   6023                      ADD    IS,3,2                                           184
002D   E035                      LDR    R,5,3                                            185
002E   E056                      LDR    R,6,5                                            186
002F   8476                      SUB    R,6,7                                            187
0030   8854                      CBL    R,4,5                                            188
                    .         ABOVE: DO                                                  189
                    @8           EQU    $                                                190
0031   C106 0041                 BRC    I,6,@9                                           191
                    .                   MP=@MP+1                                         192
0033   E205 31B5                 LDR     D,5,0+MP                                        193
0035   6015                      ADD     IS,5,1                                          194
0036   9C05 31B5                 STR     D,5,0+MP                                        195
                    .                   KK=@KK+1                                         196
0038   E205 31B6                 LDR     D,5,0+KK                                        197
003A   6015                      ADD     IS,5,1                                          198
003B   9C05 31B6                 STR     D,5,0+KK                                        199
                    .                   REPEAT LOOP1                                     200
003D   C107 000B                 BRC      I,7,@0                                         201
                    .                   END                                              202
                    .         BELOW: DO                                                  203
003F   C107 004F                 BRC      I,7,@6                                         204
                    @9           EQU    $                                                205
0041   C103 004F                 BRC    I,3,@10                                          206
                    .                   MP=@MP+1                                         207
0043   E205 31B5                 LDR     D,5,0+MP                                        208
0045   6015                      ADD     IS,5,1                                          209
0046   9C05 31B5                 STR     D,5,0+MP                                        210
                    .                   KJ=@KJ+1                                         211
0048   E205 31B7                 LDR     D,5,0+KJ                                        212
004A   6015                      ADD     IS,5,1                                          213
```

```
LOC   OBJECT CODE        CARD.IMAGE                                                              CARDNUM
004B  9C05 31B7                  STR    D,5,0+KJ                                                     214
                    .                   REPEAT LOOP1                                                 215
004D  C1C7 0008                  BRC     I,7,@0                                                      216
                    .                   END                                                          217
                    .              END                                                               218
                    @10          EQU    $                                                            219
                    @6           EQU    $                                                            220
                    .      K=@K+1                                                                    221
004F  E205 31B4                  LDR    D,5,0+K                                                      222
0051  6015                       ADD    IS,5,1                                                       223
0052  9C05 31B4                  STR    D,5,0+K                                                      224
                    .      KK=@KJ                                                                    225
0054  E205 31B7                  LDR    D,5,0+KJ                                                     226
0056  9C05 31B6                  STR    D,5,0+KK                                                     227
                    .      KJ=1                                                                      228
0058  4015                       LDR    IS,5,1                                                       229
0059  9C05 31B7                  STR    D,5,0+KJ                                                     230
                    .      TEST @K>4                                                                 231
005B  E205 31B4                  LDR    D,5,0+K                                                      232
005D  2045                       CMP    IS,5,4                                                       233
                    .          IFNOT: REPEAT LOOP1                                                   234
                    @13          EQU    $                                                            235
005E  C101 0062                  BRC    I,1,@14                                                      236
0060  C107 C00B                  BRC     I,7,@0                                                      237
                    .      END                                                                       238
                    @14          EQU    $                                                            239
                    @11          EQU    $                                                            240
                    .                                                                                241
                    ./"     COMPUTE PRI AND TEST FOR ROUND OFF ERROR.  IF ERROR EXISTS               242
                    .       ROUND UP BY 1.                                                           243
                    ."/                                                                              244
                    .                                                                                245
                    .      K=1                                                                       246
0062  4015                       LDR    IS,5,1                                                       247
0063  9C05 31B4                  STR    D,5,0+K                                                      248
                    .      PRI=0                                                                     249
0065  9C0C 30BF                  STR    D,12,0+PRI                                                   250
                    .      ITERATE 4 TIMES                                                           251
                    .      PRI=@PRI+@PI(.@K.)                                                        252
0067  4FB5                       LDR    IS,5,-5                                                      253
0068  C107 C077                  BRC     I,7,@16                                                     254
                    @15          EQU    $                                                            255
006A  E206 31B4                  LDR    D,6,0+K                                                      256
006C  E207 30BF                  LDR    D,7,0+PRI                                                    257
006E  8367 30B9                  ADD    DX,7,0+PI,6                                                  258
0070  9C07 30BF                  STR    D,7,0+PRI                                                    259
                    .      K=@K+1                                                                    260
0072  E207 31B4                  LDR    D,7,0+K                                                      261
0074  6017                       ADD    IS,7,1                                                       262
0075  9C07 31B4                  STR    D,7,0+K                                                      263
                    .      END                                                                       264
                    @16          EQU    $                                                            265
0077  C905 006A                  IBN     I,5,@15                                                     266
```

```
LOC    OBJECT CODE       CARD IMAGE                                          CARDNUM

                     .   PRITEM=@PRI
0079   E207 30BF               LDR    D,7,0+PRI
007B   9C07 31B8               STR    D,7,0+PRITEM
                     .   FRI=@PRI/(4+@MP)
007D   E207 3165               LDR    D,7,0+MP
007F   6047                    ADD    IS,7,4
0080   E203 30BF               LDR    D,3,0+PRI
0082   D8F2                    SHD    RA,2,16
0083   FC72                    DIV    R,2,7
0084   9C03 30BF               STR    D,3,0+PRI
                     .   TEST @PRITEM MOD(4+@MP)>(4+@MP)/2
0086   E202 31B5               LDR    D,2,0+MP
0088   6042                    ADD    IS,2,4
0089   E204 31B8               LDR    D,4,0+PRITEM
008B   D8F3                    SHD    RA,3,16
008C   FC23                    DIV    R,3,2
008D   E204 31B5               LDR    D,4,0+MP
008F   6044                    ADD    IS,4,4
0090   E045                    LDR    R,5,4
0091   D8F4                    SHD    RA,4,16
0092   4026                    LDR    IS,6,2
0093   FC64                    DIV    R,4,6
0094   B053                    CMP    R,3,5
                     .     IFSO: PRI=@PRI+1
                     @19       EQU    $
0095   C106 009C               BRC    I,6,@20
0097   E206 30BF               LDR    D,6,0+PRI
0099   6016                    ADD    IS,6,1
009A   9C06 30BF               STR    D,6,0+PRI
                     .   END
                     @20       EQU    $
                     @17       EQU    $
                     .
                     ./"  LOAD PARAMETERS INTO CHANNEL BUFFER  "/
                     .
                     .   DP=@NS(.@I.)
009C   E0A6                    LDR    R,6,10
009D   E367 25A0               LDR    DX,7,0+NS,6
009F   9C07 31B3               STR    D,7,0+DP
                     .   CNDX=@CNDX+6
00A1   E207 30C1               LDR    D,7,0+CNDX
00A3   6067                    ADD    IS,7,6
00A4   9C07 30C1               STR    D,7,0+CNDX
                     .   CINDX(.@I/5.)=@CNDX
00A6   E0A8                    LDR    R,8,10
00A7   D8F7                    SHD    RA,7,16
00A8   4052                    LDR    IS,2,5
00A9   FC27                    DIV    R,7,2
00AA   E202 30C1               LDR    D,2,0+CNDX
00AC   9B82 318B               STR    DX,2,0+CINDX,8
                     .   CHAN(.(@CNDX+1).)=@BUF(.(@DP+1).)
00AE   E202 30C1               LDR    D,2,0+CNDX
00B0   6012                    ADD    IS,2,1
```

```
LOC    OBJECT CODE       CARD IMAGE                                                    CARDNUM

00B1   E203 31B3                LDR    D,3,0+DP
00B3   6013                     ADD    IS,3,1
00B4   E334 2000                LDR    DX,4,0+BUF,3
00B6   9B24 30C2                STR    DX,4,0+CHAN,2
                       .  CHAN(.(@CNDX+2).)=@TMIN(.(@I/5*2).)
00B8   E204 30C1                LDR    D,4,0+CNDX
00BA   6024                     ADD    IS,4,2
00BB   E0A6                     LDR    R,6,10
00BC   D8F5                     SHD    RA,5,16
00BD   4057                     LDR    IS,7,5
00BE   FC75                     DIV    R,5,7
00BF   CD06 0002                MUL    I,6,2
00C1   E367 3068                LDR    DX,7,0+TMIN,6
00C3   9B47 30C2                STR    DX,7,0+CHAN,4
                       .  CHAN(.(@CNDX+3).)=@TMIN(.(@I/5*2+1).)
00C5   E207 30C1                LDR    D,7,0+CNDX
00C7   6037                     ADD    IS,7,3
00C8   E0A3                     LDR    R,3,10
00C9   D8F2                     SHD    RA,2,16
00CA   4054                     LDR    IS,4,5
00CB   FC42                     DIV    R,2,4
00CC   CD03 0002                MUL    I,3,2
00CE   6013                     ADD    IS,3,1
00CF   E334 3068                LDR    DX,4,0+TMIN,3
00D1   9B74 30C2                STR    DX,4,0+CHAN,7
                       .  CHAN(.(@CNDX+4).)=@BUF(.(@DP+4).)
00D3   E204 30C1                LDR    D,4,0+CNDX
00D5   6044                     ADD    IS,4,4
00D6   E205 31B3                LDR    D,5,0+DP
00D8   6045                     ADD    IS,5,4
00D9   E356 2000                LDR    DX,6,0+BUF,5
00DB   9B46 30C2                STR    DX,6,0+CHAN,4
                       .  CHAN(.(@CNDX+5).)=@PRI
00DD   E206 30C1                LDR    D,6,0+CNDX
00DF   6056                     ADD    IS,6,5
00E0   E207 30BF                LDR    D,7,0+PRI
00E2   9B67 30C2                STR    DX,7,0+CHAN,6
                       .
                       ./"  FLAG PULSE TRAIN AS STABLE AND INCREMENT CHANNEL COUNTER  "/
                       .
                       .  SSF(.(@I/5).)=TRUE
00E4   E0A8                     LDR    R,8,10
00E5   D8F7                     SHD    RA,7,16
00E6   4052                     LDR    IS,2,5
00E7   FC27                     DIV    R,7,2
00E8   4012                     LDR    IS,2,1
00E9   9B82 3163                STR    DX,2,0+SSF,8
                       .  NC=@NC+1
00EB   E202 30B8                LDR    D,2,0+NC
00ED   6012                     ADD    IS,2,1
00EE   9C02 30B8                STR    D,2,0+NC
                       .  RETURN
00F0   BF17                     BRC     R,7,1
```

```
LOC    OBJECT CODE       CARD IMAGE                                                    CARDNUM
                        .END LOOP1
                        @1        EQU     $
                        .
                        .
                        ./"  TRAIN FAILED STABILITY TEST. FLAG TRAIN AS NON STABLE  "/
                        .
                        .   NSSF(.(@I/5);)=TRUE
00F1   E0A3                       LDR    R,3,10
00F2   D8F2                       SHD    RA,2,16
00F3   4054                       LDR    IS,4,5
00F4   FC42                       DIV    R,2,4
00F5   4014                       LDR    IS,4,1
00F6   9B34 313B                  STR    DX,4,0+NSSF,3
                        .END PRICCMP
00F8   BF17                       BRC     R,7,1
                        .
                        .
                        .PROCEDURE MAIN
                                  ENTRY MAIN
                        .
                        ./"  DESCRIPTION OF VARIABLES
                        .
                        .    BRNGF   RCVR ANTENNA BEARING AT TOA OF FIRST PULSE OF A SIGNAL
                        .    BRNGL   RCVR ANTENNA BEARING AT TOA OF LAST PULSE OF A TRAIN
                        .    CON1    DOUBLE PRECISION TIME CALCULATION VARIABLE
                        .    CON2    DOUBLE PRECISION TIME CALCULATION VARIABLE
                        .    II,JJ   INDICES
                        .    SCANTM  RCVR ANTENNA SCAN TIME FOR ONE BEAMWIDTH
                        .    TOAL    LOWER HALF DOUBLE PRECISION TIME OF ARRIVAL
                        .    TOAU    UPPER HALF DOUBLE PRECISION TIME OF ARRIVAL
                        .    TMAX    EXPECTED TIME OF LAST PULSE INTERCEPT
                        .    TMIN    DOUBLE PRECISION TIME OF ARRIVAL OF FIRST PULSE
                        .    TTEM    DOUBLE PRECISION TIME CALCULATION VARIABLE
                        .    TTEST   DOUBLE PRECISION TIME CALCULATION VARIABLE
                        ."/
                        .
                        .DECLARE(BRNGF,BRNGL(39),SCANTM(1),II,JJ,STR)RAM
                                  ORG    RAM
31B9                    BRNGF     DS     1
31BA                    BRNGL     DS     40
31E2                    SCANTM    DS     2
31E4                    II        DS     1
31E5                    JJ        DS     1
31E6                    STR       DS     1
                        .DECLARE(TOAL,TOAU,TMAX(79),TTEST(1),TTEM(1))RAM
31E7                    TOAL      DS     1
31E8                    TOAU      DS     1
31E9                    TMAX      DS     80
3239                    TTEST     DS     2
323B                    TTEM      DS     2
                        .DECLARE(CON1(1),CON2(1))RAM
323D                    CON1      DS     2
323F                    CON2      DS     2
```

```
LOC   OBJECT CODE    CARD IMAGE

                     .
                     ./"  INITIALIZE VARIABLES  "/
                     .
                     .    CNDX=-6
                                ORG    ROM
                     MAIN       EQU    $
00F9  400C                      LDR    IS,12,0
00FA  4FA2                      LDR    IS,2,-6
00FB  9C02 30C1                 STR    D,2,0+CNDX
                     .    SCANTM=1120
00FD  E102 0460                 LDR    I,2,1120
00FF  9C02 31E2                 STR    D,2,0+SCANTM
                     .    I=0
0101  E0CA                      LDR    R,10,12
                     .    BP=0
0102  E0CB                      LDR    R,11,12
                     .    NC=1
0103  4012                      LDR    IS,2,1
0104  9C02 30B8                 STR    D,2,0+NC
                     .
                     ./"  SET NEW SIGNAL POINTER FILE TO NEGATIVE VALUES  "/
                     .
                     .    ITERATE 200 TIMES
                     .    NS(.@I.)=-1
0106  E102 FF37                 LDR    I,2,-201
0108  C107 0111                 BRC    I,7,@22
                     @21        EQU    $
010A  E0A3                      LDR    R,3,10
010B  4FF4                      LDR    IS,4,-1
010C  9B34 2FA0                 STR    DX,4,0+NS,3
                     .    I=@I+1
010E  E0A4                      LDR    R,4,10
010F  6014                      ADD    IS,4,1
0110  E04A                      LDR    R,10,4
                     .    END
                     @22        EQU    $
0111  C502 010A                 IBN    I,2,@21
                     .    I=0
0113  E0CA                      LDR    R,10,12
                     .    NS(.@I.)=0
0114  E0A4                      LDR    R,4,10
0115  9B4C 2FA0                 STR    DX,12,0+NS,4
                     .
                     ./"  LOAD INPUT BUFFER FROM CARDS  "/
                     .
                     .    READ_DEC(BUF,BUF+3999)
0117  FE01 011A                 SIM    1,@23
0119  0017                      BRC    IS,7,@24
                     @23        EQU    $
011A  2000                      DC     0+BUF
011B  2F9F                      DC     3999+BUF
                     @24        EQU    $
                     .
```

```
LOC   OBJECT CODE     CARD IMAGE                                                      CARDNUM
                    .MAINLP:
                    .%  DO
                    @25          EQU    $
                    .
                    ./"  LOAD DP TOA OF FIRST PULSE OF SIGNAL FROM INPUT BUFFER AND
                    .    COMPUTE EXPECTED TIME OF ARRIVAL OF LAST PULSE BASED ON RCVR
                    .    ANTENNA SCAN TIME FOR ONE ANTENNA BEAMWIDTH  "/
                    .
                    .    TOAL=@BUF(.(@BP+2).)
011C  E0B4                       LDR    R,4,11
011D  6024                       ADD    IS,4,2
011E  E345 2000                  LDR    DX,5,0+BUF,4
0120  9C05 31E7                  STR    D,5,0+TOAL
                    .    TOAU=@BUF(.(@BP+3).)
0122  E0B5                       LDR    R,5,11
0123  6035                       ADD    IS,5,3
0124  E356 2000                  LDR    DX,6,0+BUF,5
0126  9C06 31E8                  STR    D,6,0+TOAU
                    .    CODE
0128  BC1E 31E7                  LDRM   D,14,TOAL,2
012A  A90E 31E2                  DADD   D,14,SCANTM
012C  9C1E 323B                  STRM   D,14,TTEM,2
                    .    END
                    .    STR=@I/5*2
012E  E0A7                       LDR    R,7,10
012F  D8F6                       SHD    RA,6,16
0130  4058                       LDR    IS,8,5
0131  FC86                       DIV    R,6,8
0132  CD07 0002                  MUL    I,7,2
0134  9C07 31E6                  STR    D,7,0+STR
                    .    TMAX(.@STR.)=@TTEM
0136  E208 31E6                  LDR    D,8,0+STR
0138  E202 323B                  LDR    D,2,0+TTEM
013A  9B82 31E9                  STR    DX,2,0+TMAX,8
                    .    TMAX(.(@STR+1).)=@TTEM(.1.)
013C  E202 31E6                  LDR    D,2,0+STR
013E  6012                       ADD    IS,2,1
013F  E203 323C                  LDR    D,3,1+TTEM
0141  9B23 31E9                  STR    DX,3,0+TMAX,2
                    .    TMIN(.@STR.)=@TOAL
0143  E203 31E6                  LDR    D,3,0+STR
0145  E204 31E7                  LDR    D,4,0+TOAL
0147  9B34 3068                  STR    DX,4,0+TMIN,3
                    .    TMIN(.(@STR+1).)=@TOAU
0149  E204 31E6                  LDR    D,4,0+STR
014B  6014                       ADD    IS,4,1
014C  E205 31E8                  LDR    D,5,0+TOAU
014E  9B45 3068                  STR    DX,5,0+TMIN,4
                    .COMPUTE:
                    .%  DO
                    @27          EQU    $
                    .
                    ./"  INCREMENT BUFFER POINTER AND CHECK AT BUFFER LIMIT.  IFSO, GO
```

```
LCC    CBJECT CODE     CARD IMAGE                                                            CARDNUM
                       .
                       ."/   COMPRESS RESIDUAL PULSES WITHIN INPUT BUFFER FOR FUTHER PROCESSING
                       .
                       .     BP=@BP+5
0150   E0B5                        LDR     R,5,11
0151   6055                        ADD     I6,5,5
0152   E05B                        LDR     R,11,5
                       .     TEST @BP>3999
0153   E0B5                        LDR     R,5,11
0154   B105 0F9F                   CMP     I,5,3999
                       .        IFSO: LEAVE MAINLP
                       @31                EQU     $
0156   C106 015A                   BRC     I,6,@32
0158   C107 02CD                   BRC      I,7,@26
                       .     END
                       @32         EQU     $
                       @29         EQU     $
                       .
                       ./"   IF NO SIGNAL IN BUFFER INDEXED BY BUFFER POINTER INCREMENT
                       .     POINTER AND REPEAT COMPUTE  "/
                       .
                       .     TEST @BUF(.@BP.)
015A   E0B5                        LDR     R,5,11
015B   E356 2000                   LDR     DX,6,0+BUF,5
                       .        ZERO: REPEAT COMPUTE
                       @35         EQU     $
015D   C105 0161                   BRC     I,5,@36
015F   C107 0150                   BRC      I,7,@27
                       .     END
                       @36         EQU     $
                       @33         EQU     $
                       .     I=0
0161   E0CA                        LDR     R,10,12
                       .     J=0
0162   E0C9                        LDR     R,9,12
                       .SORT:
                       .%    DO
                       @37         EQU     $
                       .
                       ./"   IF AT END OF NEW SIGNAL POINTER FILE, LEAVE SORT  "/
                       .
                       .     TEST @I>195
0163   E0A6                        LDR     R,6,10
0164   B106 00C3                   CMP     I,6,195
                       .        IFSO: LEAVE SORT
                       @41         EQU     $
0166   C106 016A                   BRC     I,6,@42
0168   C107 02B3                   BRC      I,7,@38
                       .     END
                       @42         EQU     $
                       @39         EQU     $
                       .
                       ./"   IF NEW SIGNAL POINTER FILE INDEXED BY I IS EMPTY INDEX AND
```

```
LOC   OBJECT CODE      CARD IMAGE                                                                   CARDNUM
                    .   CHECK NEXT LOCATION  "/                                                      585
                    .                                                                                586
                    .  TEST @NS(.@I.)                                                                587
016A  E0A6                   LDR    R,6,10                                                           588
016B  E367 2FA0.             LDR    DX,7,0+NS,6                                                      589
                    .      NEG: DO                                                                   590
                    @45      EQU    $                                                                591
016D  C103 0174              BRC    I,3,@46                                                          592
                    .         I=@I+5                                                                 593
016F  E0A7                   LDR    R,7,10                                                           594
0170  6057                   ADD    IS,7,5                                                           595
0171  E07A                   LDR    R,10,7                                                           596
                    .         REPEAT SORT                                                            597
0172  C107 0163              BRC       I,7,@37                                                       598
                    .       END                                                                      599
                    .  END                                                                           600
                    @46      EQU    $                                                                601
                    @43      EQU    $                                                                602
                    .                                                                                603
                    ./"  COMPARE TOA OF BUFFER SIGNAL WITH TMAX OF SIGNAL FOUND IN NEW               604
                    .    SIGNAL POINTER FILE.  IF TOA OF BUFFER SIGNAL IS GREATER THAN               605
                    .    OR EQUAL TO TMAX, TEST FOR STABLE SIGNAL IN NS POINTER FILE.                606
                    .    IF  SIGNAL HAS BEEN CLASSIFIED STABLE, RESET ACTIVE CHANNEL                 607
                    .    COUNTER AND SSF FLAG, ZERO FIRST FIVE PULSES FROM THIS SIGNAL               608
                    .    IN INPUT BUFFER AND COMPUTE EMITTER BEARING USING FIRST AND                 609
                    .    DOA OF SIGNAL "/                                                            610
                    .                                                                                611
                    .  CON1=@BUF(.(@BP+2).)                                                          612
0174  E0B7                   LDR    R,7,11                                                           613
0175  6C27                   ADD    IS,7,2                                                           614
0176  E378 2000              LDR    DX,8,0+BUF,7                                                     615
0178  9C08 323D              STR    D,8,0+CON1                                                       616
                    .  CON1(.1.)=@BUF(.(@BP+3).)                                                     617
017A  E0B8                   LDR    R,8,11                                                           618
017B  6038                   ADD    IS,8,3                                                           619
017C  E382 2000              LDR    DX,2,0+BUF,8                                                     620
017E  9C02 323E              STR    D,2,1+CON1                                                       621
                    .  CON2=@TMAX(.(@I/5*2).)                                                        622
0180  E0A3                   LDR    R,3,10                                                           623
0181  D8F2                   SHD    RA,2,16                                                          624
0182  4054                   LDR    IS,4,5                                                           625
0183  FC42                   DIV    R,2,4                                                            626
0184  CD03 0002              MUL    I,3,2                                                            627
0186  E334 31E9              LDR    DX,4,0+TMAX,3                                                    628
0188  9C04 323F              STR    D,4,0+CON2                                                       629
                    .  CON2(.1.)=@TMAX(.(@I/5*2+1).)                                                 630
018A  E0A5                   LDR    R,5,10                                                           631
018B  D8F4                   SHD    RA,4,16                                                          632
018C  4056                   LDR    IS,6,5                                                           633
018D  FC64                   DIV    R,4,6                                                            634
018E  CD05 0002              MUL    I,5,2                                                            635
0190  6015                   ADD    IS,5,1                                                           636
0191  E356 31E9              LDR    DX,6,0+TMAX,5                                                    637
```

```
LOC   OBJECT CODE      CARD IMAGE                                                        CARDNUM
0193  9C06 3240              STR    D,6,1+CON2                                              638
                   .   CODE                                                                 639
0195  BC1E 323D              LDRM   D,14,CON1,2                                             640
0197  BC17 323F              LDRM   D,7,CON2,2                                              641
0199  AAE7                   DSUB   R,7,14                                                  642
019A  9C17 3239              STRM   D,7,TTEST,2                                             643
                   .   END                                                                  644
                   .   TEST @TTEST                                                          645
019C  E206 3239              LDR    D,6,0+TTEST                                             646
                   .      NEG,ZERO: DO                                                      647
                   @49       EQU    $                                                       648
019E  C101 0208              BRC    I,1,@50                                                 649
                   .           TEST @SSF(.(@I/5).)                                          650
01A0  E0A7                   LDR    R,7,10                                                  651
01A1  D8F6                   SHD    RA,6,16                                                 652
01A2  4058                   LDR    IS,8,5                                                  653
01A3  FC86                   DIV    R,6,8                                                   654
01A4  E378 3163              LDR    DX,8,0+SSF,7                                            655
                   .                POS: DO                                                 656
                   @53       EQU    $                                                       657
01A6  C106 01EE              BRC    I,6,@54                                                 658
                   .                      NC=@NC-1                                          659
01A8  E208 30B8              LDR    D,8,0+NC                                                660
01AA  6FF8                   ADD    IS,8,-1                                                 661
01AB  9C08 30B8              STR    D,8,0+NC                                                662
                   .                      SSF(.(@I/5).)=FALSE                               663
01AD  E0A3                   LDR    R,3,10                                                  664
01AE  D8F2                   SHD    RA,2,16                                                 665
01AF  4054                   LDR    IS,4,5                                                  666
01B0  FC42                   DIV    R,2,4                                                   667
01B1  9B3C 3163              STR    DX,12,0+SSF,3                                           668
                   .                      BRNGF=@BUF(.(@NS(.@I.)).)                         669
01B3  E0A4                   LDR    R,4,10                                                  670
01B4  E345 2FA0              LDR    DX,5,0+NS,4                                             671
01B6  E356 2000              LDR    DX,6,0+BUF,5                                            672
01B8  9C06 31B9              STR    D,6,0+BRNGF                                             673
                   .                        II=0                                            674
01BA  9C0C 31E4              STR    D,12,0+II                                               675
                   .                      ITERATE 5 TIMES                                   676
                   .                        JJ=0                                            677
01BC  4FA6                   LDR    IS,6,-6                                                 678
01BD  C107 01D9              BRC    I,7,@56                                                 679
                   @55       EQU    $                                                       680
01BF  9C0C 31E5              STR    D,12,0+JJ                                               681
                   .                        ITERATE 5 TIMES                                 682
                   .                          BUF(.(@NS(.(@I+@II).)+@JJ).)=0                683
01C1  4FA7                   LDR    IS,7,-6                                                 684
01C2  C107 01D2              BRC    I,7,@58                                                 685
                   @57       EQU    $                                                       686
01C4  E0A8                   LDR    R,8,10                                                  687
01C5  8208 31E4              ADD    D,8,0+II                                                688
01C7  E382 2FA0              LDR    DX,2,0+NS,8                                             689
01C9  8202 31E5              ADD    D,2,0+JJ                                                690
```

```
LOC   OBJECT CODE      CARD IMAGE                                                                   CARDNUM
01CB  9B2C 2000              STR    DX,12,0+BUF,2                                                      691
                     .                          JJ=@JJ+1                                                692
01CD  E202 31E5              LDR    D,2,0+JJ                                                           693
01CF  6012                   ADD    IS,2,1                                                             694
01D0  9C02 31E5              STR    D,2,0+JJ                                                           695
                     .                     END                                                          696
                     @58     EQU    $                                                                  697
01D2  C907 01C4              IBN     I,7,@57                                                           698
                     .                     II=@II+1                                                     699
01D4  E202 31E4              LDR    D,2,0+II                                                           700
01D6  6012                   ADD    IS,2,1                                                             701
01D7  9C02 31E4              STR    D,2,0+II                                                           702
                     .                 END                                                              703
                     @56     EQU    $                                                                  704
01D9  C906 01BF              IBN     I,6,@55                                                           705
                     .                CHAN(.@CINDX(.@I/5.).)=(@BRNGF+@BRNGL(.@I/5.))/2                  706
01DB  E0A3                   LDR    R,3,10                                                             707
01DC  D8F2                   SHD    RA,2,16                                                            708
01DD  4054                   LDR    IS,4,5                                                             709
01DE  FC42                   DIV    R,2,4                                                              710
01DF  E334 318B              LDR    DX,4,0+CINDX,3                                                     711
01E1  E0A6                   LDR    R,6,10                                                             712
01E2  D8F5                   SHD    RA,5,16                                                            713
01E3  4057                   LDR    IS,7,5                                                             714
01E4  FC75                   DIV    R,5,7                                                              715
01E5  E207 31B9              LDR    D,7,0+BRNGF                                                        716
01E7  8367 31BA              ADD    DX,7,0+BRNGL,6                                                     717
01E9  D8F6                   SHD    RA,6,16                                                            718
01EA  4028                   LDR    IS,8,2                                                             719
01EB  FC86                   DIV    R,6,8                                                              720
01EC  9B47 30C2              STR    DX,7,0+CHAN,4                                                      721
                     .                 END                                                              722
                     .        END                                                                       723
                     @54     EQU    $                                                                  724
                     @51     EQU    $                                                                  725
                     .                                                                                  726
                     ./"  IF NS POINTER FILE SIGNAL HAS BEEN CLASSIFIED AS NON STABLE                  727
                     .    RESET NSSF FLAG  "/                                                          728
                     .                                                                                  729
                     .        NSSF(.(@I/5).)=FALSE                                                     730
01EE  E0A3                   LDR    R,3,10                                                             731
01EF  D8F2                   SHD    RA,2,16                                                            732
01F0  4054                   LDR    IS,4,5                                                             733
01F1  FC42                   DIV    R,2,4                                                              734
01F2  9B3C 313B              STR    DX,12,0+NSSF,3                                                     735
                     .                                                                                  736
                     ./"  CLEAR THE SIGNAL FROM NS POINTER FILE  "/                                    737
                     .                                                                                  738
                     .        II=0                                                                     739
01F4  9C0C 31E4              STR    D,12,0+II                                                          740
                     .        ITERATE 5 TIMES                                                          741
                     .        NS(.(@I+@II).)=-1                                                        742
01F6  4FA4                   LDR    IS,4,-6                                                            743
```

```
LOC    OBJECT CODE      CARD IMAGE                                                    CARDNUM
01F7   C107 0204                BRC     I,7,@60
                   @59          EQU     $
01F9   E0A5                     LDR     R,5,10
01FA   8205 31E4                ADD     D,5,0+II
01FC   4FF6                     LDR     IS,6,-1
01FD   9B56 2FA0                STR     DX,6,0+NS,5
                   .             II=@II+1
01FF   E206 31E4                LDR     D,6,0+II
0201   6016                     ADD     IS,6,1
0202   9C06 31E4                STR     D,6,0+II
                   .             END
                   @60          EQU     $
0204   C904 01F9                IBN      I,4,@59
                   .             I=@I+5
0206   E0A6                     LDR     R,6,10
0207   6056                     ADD     IS,6,5
0208   E06A                     LDR     R,10,6
                   .             REPEAT SORT
0209   C107 0163                BRC      I,7,@37
                   .          END
                   .     END
                   @50          EQU     $
                   @47          EQU     $
                   .     TEST @BUF(.(@BP+1).)==@BUF(.(@NS(.@I.)+1).)
020B   E0B6                     LDR     R,6,11
020C   6016                     ADD     IS,6,1
020D   E0A7                     LDR     R,7,10
020E   E378 2FA0                LDR     DX,8,0+NS,7
0210   6018                     ADD     IS,8,1
0211   E362 2000                LDR     DX,2,0+BUF,6
0213   B382 2000                CMP     DX,2,0+BUF,8
                   .        IFNOT: DO
                   @63          EQU     $
0215   C102 021C                BRC     I,2,@64
                   .             I=@I+5
0217   E0A2                     LDR     R,2,10
0218   6052                     ADD     IS,2,5
0219   E02A                     LDR     R,10,2
                   .             REPEAT SORT
021A   C107 0163                BRC      I,7,@37
                   .             END
                   .     END
                   @64          EQU     $
                   @61          EQU     $
                   .
                   ./"    TEST FOR MATCH BETWEEN BUFFER SIGNAL FREQ AND PW AND NS SIGNAL
                   .      POINTER FILE FREQ AND PW.  IF BOTH DO NOT MATCH REPEAT SORT.
                   .      IF MATCH OCCURS AND SIGNAL HAS BEEN CLASSIFIED AS STABLE,
                   .      ZERO BUFFER SIGNAL AND REPEAT COMPUTE.  IF SIGNAL HAS BEEN
                   .      CLASSIFIED AS NON STABLE, REPEAT COMPUTE.  "/
                   .
                   .     TEST @BUF(.(@BP+4).)==@BUF(.(@NS(.@I.)+4).)
021C   E0B2                     LDR     R,2,11
```

```
LOC   OBJECT CODE      CARD IMAGE                                         CARDNUM
021D  6042                     ADD    IS,2,4                                 797
021E  E0A3                     LDR    R,3,10                                 798
021F  E334 2FA0                LDR    DX,4,0+NS,3                            799
0221  6044                     ADD    IS,4,4                                 800
0222  E325 2000                LDR    DX,5,0+BUF,2                           801
0224  B345 2000                CMP    DX,5,0+BUF,4                           802
                  .          IFNOT: DO                                       803
                  @67          EQU    $                                      804
0226  C102 022D                BRC    I,2,@68                                805
                  .             I=@I+5                                       806
0228  E0A5                     LDR    R,5,10                                 807
0229  6055                     ADD    IS,5,5                                 808
022A  E05A                     LDR    R,10,5                                 809
                  .             REPEAT SORT                                  810
022B  C107 0163                BRC    I,7,@37                                811
                  .             END                                          812
                  .      END                                                 813
                  @68          EQU    $                                      814
                  @65          EQU    $                                      815
                  .      TEST @SSF(.(@I/5).)                                 816
022D  E0A6                     LDR    R,6,10                                 817
022E  D8F5                     SHD    RA,5,16                                818
022F  4057                     LDR    IS,7,5                                 819
0230  FC75                     DIV    R,5,7                                  820
0231  E367 3163                LDR    DX,7,0+SSF,6                           821
                  .          NONZERO: DO                                     822
                  @71          EQU    $                                      823
0233  C102 0251                BRC    I,2,@72                                824
                  .             BRNGL(.@I/5.)=@BUF(.@BP.)                    825
0235  E0A8                     LDR    R,8,10                                 826
0236  D8F7                     SHD    RA,7,16                                827
0237  4052                     LDR    IS,2,5                                 828
0238  FC27                     DIV    R,7,2                                  829
0239  E082                     LDR    R,2,11                                 830
023A  E323 2000                LDR    DX,3,0+BUF,2                           831
023C  9383 31BA                STR    DX,3,0+BRNGL,8                         832
                  .             II=0                                         833
023E  9C0C 31E4                STR    D,12,0+II                              834
                  .             ITERATE 5 TIMES                              835
                  .             BUF(.(@BP+@II).)=0                           836
0240  4FA3                     LDR    IS,3,-6                                837
0241  C107 0240                BRC    I,7,@74                                838
                  @73          EQU    $                                      839
0243  E0B4                     LDR    R,4,11                                 840
0244  8204 31E4                ADD    D,4,0+II                               841
0246  984C 2000                STR    DX,12,0+BUF,4                          842
                  .             II=@II+1                                     843
0248  E204 31E4                LDR    D,4,0+II                               844
024A  6014                     ADD    IS,4,1                                 845
024B  9C04 31E4                STR    D,4,0+II                               846
                  .             END                                          847
                  @74          EQU    $                                      848
024D  C903 0243                IBN    I,3,@73                                849
```

```
LOC    OBJECT CODE      CARD IMAGE                                                        CARDNUM
                      .        REPEAT COMPUTE                                                850
024F   C107 0150                  BRC     I,7,@27                                            851
                      .        END                                                           852
                      .     END                                                              853
                      @72         EQU    $                                                   854
                      @69         EQU    $                                                   855
                      .     TEST @NSSF(.(@I/5).)==TRUE                                       856
0251   E0A5                       LDR    R,5,10                                              857
0252   D8F4                       SHD    RA,4,16                                             858
0253   4056                       LDR    IS,6,5                                              859
0254   FC64                       DIV    R,4,6                                               860
0255   E356 313B                  LDR    DX,6,0+NSSF,5                                       861
0257   2016                       CMP    IS,6,1                                              862
                      .        IFSU: REPEAT COMPUTE                                          863
                      @77         EQU    $                                                   864
0258   C105 025C                  BRC    I,5,@78                                             865
025A   C107 0150                  BRC     I,7,@27                                            866
                      .     END                                                              867
                      @78         EQU    $                                                   868
                      @75         EQU    $                                                   869
                      .                                                                      870
                      ./"   IF SIGNAL HAS NOT BEEN CLASSIFIED, STORE BUFFER POINTER IN       871
                      .     NS POINTER FILE. IF THIS IS THE FIFTH PULSE ASSOCIATED WITH      872
                      .     THIS TRAIN, COMPUTE TIME INTERVALS BETWEEN PULSES AND CALL       873
                      .     FOR PRI COMPUTATION.    "/                                       874
                      .                                                                      875
                      .ITERATE 3 TIMES                                                       876
                      .     J=@J+1                                                           877
025C   4FC6                       LDR    IS,6,-4                                             878
025D   C107 026E                  BRC     I,7,@80                                            879
                      @79         EQU    $                                                   880
025F   E097                       LDR    R,7,9                                               881
0260   6017                       ADD    IS,7,1                                              882
0261   E079                       LDR    R,9,7                                               883
                      .     TEST @NS(.(@I+@J).)                                              884
0262   E0A7                       LDR    R,7,10                                              885
0263   8097                       ADD    R,7,9                                               886
0264   E378 2FA0                  LDR    DX,8,0+NS,7                                         887
                      .        NEG: DO                                                       888
                      @83         EQU    $                                                   889
0266   C103 026E                  BRC    I,3,@84                                             890
                      .            NS(.(@I+@J).)=@BP                                         891
0268   E0A8                       LDR    R,8,10                                              892
0269   8098                       ADD    R,8,9                                               893
026A   9B8B 2FA0                  STR    DX,11,0+NS,8                                        894
                      .            REPEAT COMPUTE                                            895
026C   C107 0150                  BRC     I,7,@27                                            896
                      .            END                                                       897
                      .     END                                                              898
                      @84         EQU    $                                                   899
                      @81         EQU    $                                                   900
                      .     END                                                              901
                      @80         EQU    $                                                   902
```

```
LOC   OBJECT CODE       CARD IMAGE                                                          CARDNUM

026E  C906 025F                IBN     I,6,@79                                                  903
                    .   NS(.(@I+4).)=@BP                                                        904
0270  E0A8                     LDR     R,8,10                                                   905
0271  6048                     ADD     IS,8,4                                                   906
0272  9B8B 2FA0                STR     DX,11,0+NS,8                                             907
                    .   TEST @NC==10                                                            908
0274  E208 30B8                LDR     D,8,0+NC                                                 909
0276  20A8                     CMP     IS,8,10                                                  910
                    .     IFSO: DO                                                              911
                    @87        EQU     $                                                        912
0277  C105 0282                BRC     I,5,@88                                                  913
                    .           NSSF(.(@I/5).)=TRUE                                             914
0279  E0A3                     LDR     R,3,10                                                   915
027A  D8F2                     SHD     RA,2,16                                                  916
027B  4054                     LDR     IS,4,5                                                   917
027C  FC42                     DIV     R,2,4                                                    918
027D  4014                     LDR     IS,4,1                                                   919
027E  9B34 313B                STR     DX,4,0+NSSF,3                                            920
                    .           REPEAT COMPUTE                                                  921
0280  C107 0150                BRC     I,7,@27                                                  922
                    .           END                                                             923
                    .   END                                                                     924
                    @88        EQU     $                                                        925
                    @85        EQU     $                                                        926
                    .   J=1                                                                     927
0282  4019                     LDR     IS,9,1                                                   928
                    .   ITERATE 4 TIMES                                                         929
                    .   PI(.@J.)=@BUF(.(@NS(.(@I+@J).)+2).)-@BUF(.(@NS(.(@I+@J-1).)+2).)        930
0283  4FB4                     LDR     IS,4,-5                                                  931
0284  C107 02A8                BRC     I,7,@90                                                  932
                    @89        EQU     $                                                        933
0286  E095                     LDR     R,5,9                                                    934
0287  E0A6                     LDR     R,6,10                                                   935
0288  6096                     ADD     R,6,9                                                    936
0289  E367 2FA0                LDR     DX,7,0+NS,6                                              937
028B  6027                     ADD     IS,7,2                                                   938
028C  E0A8                     LDR     R,8,10                                                   939
028D  8098                     ADD     R,8,9                                                    940
028E  6FF8                     ADD     IS,8,-1                                                  941
028F  E382 2FA0                LDR     DX,2,0+NS,8                                              942
0291  6022                     ADD     IS,2,2                                                   943
0292  E373 2000                LDR     DX,3,0+BUF,7                                             944
0294  8723 2000                SUB     DX,3,0+BUF,2                                             945
0296  9B53 30B9                STR     DX,3,0+PI,5                                              946
                    .   TEST @PI(.@J.)                                                          947
0298  E093                     LDR     R,3,9                                                    948
0299  E335 30B9                LDR     DX,5,0+PI,3                                              949
                    .     NEG:   PI(.@J.)=@PI(.@J.)+32767                                       950
                    @93        EQU     $                                                        951
029B  C103 02A5                BRC     I,3,@94                                                  952
029D  E095                     LDR     R,5,9                                                    953
029E  E096                     LDR     R,6,9                                                    954
029F  E367 30B9                LDR     DX,7,0+PI,6                                              955
```

```
LOC   OBJECT CODE      CARD IMAGE                                   CARDNUM
02A1  8107 7FFF               ADD    I,7,32767                        956
02A3  9B57 30B9               STR    DX,7,0+PI,5                      957
                       .   END                                        958
                       @94        EQU    $                            959
                       @91        EQU    $                            960
                       .   J=@J+1                                     961
02A5  E097                    LDR    R,7,9                            962
02A6  6017                    ADD    IS,7,1                           963
02A7  E079                    LDR    R,9,7                            964
                       .   END                                        965
                       @90        EQU    $                            966
02A8  C904 0286               IBN      I,4,@89                        967
                       .   CALL PRICOMP()                             968
02AA  6010                    ADD    IS,0,1                           969
02AB  6DA0                    STK    DR,0,11                          970
02AC  ED01 0000               BAL    I,1,0+PRICOMP                    971
02AE  8EA0                    RET    DR,0,11                          972
02AF  6FF0                    ADD    IS,0,-1                          973
02B0  5C07                    LDR    RX,7,0,12                        974
                       .   REPEAT COMPUTE                             975
02B1  C107 0150               BRC      I,7,@27                        976
                       .END SORT                                      977
                       @38        EQU    $                            978
                       .   I=0                                        979
02B3  E0CA                    LDR    R,10,12                          980
                       .   CYCLE                                      981
                       .   TEST @NS(.@I.)                             982
                       @95        EQU    $                            983
02B4  E0A7                    LDR    R,7,10                           984
02B5  E378 2FA0               LDR    DX,8,0+NS,7                      985
                       .      NEG: DO                                 986
                       @98        EQU    $                            987
02B7  C103 02BE               BRC    I,3,@99                          988
                       .          NS(.@I.)=@BP                        989
02B9  E0A8                    LDR    R,8,10                           990
02BA  9B8B 2FA0               STR    DX,11,0+NS,8                     991
                       .          REPEAT MAINLP                       992
02BC  C107 011C               BRC      I,7,@25                        993
                       .          END                                 994
                       .   END                                        995
                       @99        EQU    $                            996
                       @96        EQU    $                            997
                       .   TEST(I=@I+5)>195                           998
02BE  E0A8                    LDR    R,8,10                           999
02BF  6058                    ADD    IS,8,5                          1000
02C0  E08A                    LDR    R,10,8                          1001
02C1  B108 00C3               CMP    I,8,195                         1002
                       .       IFSO: DO                              1003
                       @102       EQU    $                           1004
02C3  C106 02CB               BRC    I,6,@103                        1005
                       .              DUMP_DEC(NS,NS+195)            1006
02C5  FE0D 02C8               SIM    13,@104                         1007
02C7  0017                    BRC    IS,7,@105                       1008
```

```
LOC   OBJECT CODE     CARD IMAGE                                                                    CARDNUM
                      @104      EQU    $                                                            1009
02C8  2FA0                      DC     0+NS                                                         1010
02C9  3063                      DC     195+NS                                                       1011
                      @105      EQU    $                                                            1012
                      .               RETURN                                                        1013
02CA  BF17                      BRC      R,7,1                                                      1014
                      .               END                                                           1015
                      .    END                                                                      1016
                      @103      EQU    $                                                            1017
                      @100      EQU    $                                                            1018
                      .    END                                                                      1019
02C6  C107 02B4                 BRC      I,7,@95                                                    1020
                      .END COMPUTE                                                                  1021
                      @28       EQU    $                                                            1022
                      .END MAINLP                                                                   1023
                      @26       EQU    $                                                            1024
                      .                                                                             1025
                      ./"  ROUTINE BELOW COMPRESSES RESIDUAL PULSES WITHIN THE INPUT                1026
                      .    BUFFER. AN ATAC ASSEMBLY LANGUAGE PROGRAM TO PROCESS THIS               1027
                      .    RESIDUAL MAY BE INCORPORATED BETWEEN THE END AND END MAIN               1028
                      .    STATEMENTS UTILIZING THE CODE COLLECTION HEAD.  "/                       1029
                      .                                                                             1030
                      .    BP=0                                                                     1031
02CD  E0CB                      LDR    R,11,12                                                      1032
                      .    ZEROCK:                                                                  1033
                      .%      CYCLE                                                                 1034
                      @106      EQU    $                                                            1035
                      .    TEST @BUF(.@BP+1.)                                                       1036
                      @108      EQU    $                                                            1037
02CE  E088                      LDR    R,8,11                                                       1038
02CF  6018                      ADD    IS,8,1                                                       1039
02D0  E382 2000                 LDR    DX,2,0+BUF,8                                                 1040
                      .       ZERO:  DO                                                             1041
                      @111      EQU    $                                                            1042
02D2  C105 02D8                 BRC    I,5,@112                                                     1043
                      .          DP=@BP                                                             1044
02D4  9C08 3183                 STR    D,11,0+DP                                                    1045
                      .          LEAVE ZEROCK                                                       1046
02D6  C107 02DD                 BRC      I,7,@107                                                   1047
                      .       END                                                                   1048
                      .    END                                                                      1049
                      @112      EQU    $                                                            1050
                      @109      EQU    $                                                            1051
                      .    BP=@BP+5                                                                 1052
02D8  E082                      LDR    R,2,11                                                       1053
02D9  6052                      ADD    IS,2,5                                                       1054
02DA  E02B                      LDR    R,11,2                                                       1055
                      .END ZERCCK                                                                   1056
02DB  C107 02CE                 BRC      I,7,@108                                                   1057
                      @107      EQU    $                                                            1058
                      .    CYCLE                                                                    1059
                      .         BP=@BP+5                                                            1060
                      @113      EQU    $                                                            1061
```

```
LOC   OBJECT CODE      CARD IMAGE                                                              CARDNUM

02DD  E0B2                        LDR    R,2,11                                                1062
02DE  6052                        ADD    IS,2,5                                                1063
02DF  E02B                        LDR    R,11,2                                                1064
                        .       TEST @BP>3995                                                  1065
02E0  E0B2                        LDR    R,2,11                                                1066
02E1  B102 0F9B                   CMP    I,2,3995                                              1067
                        .          IFSO: DO                                                    1068
                        @116      EQU    $                                                     1069
02E3  C106 02E6                   BRC    I,6,@117                                              1070
                        .                 RETURN                                               1071
02E5  BF17                        BRC     R,7,1                                                1072
                        .                 END                                                  1073
                        .       END                                                            1074
                        @117      EQU    $                                                     1075
                        @114      EQU    $                                                     1076
                        .          TEST @BUF(.(@BP+1).)==0                                     1077
02E6  E0B2                        LDR    R,2,11                                                1078
02E7  6012                        ADD    IS,2,1                                                1079
02E8  E323 2000                   LDR    DX,3,0+BUF,2                                          1080
02EA  2003                        CMP    IS,3,0                                                1081
                        .                IFNOT: DO                                             1082
                        @120      EQU    $                                                     1083
02EB  C102 0308                   BRC    I,2,@121                                              1084
                        .                   I=0                                                1085
02ED  E0CA                        LDR    R,10,12                                               1086
                        .                     ITERATE 5 TIMES                                  1087
                        .                        BUF(.(@DP+@I).)=@BUF(.(@BP+@I).)              1088
02EE  4FA3                        LDR    IS,3,-6                                               1089
02EF  C107 0301                   BRC     I,7,@123                                             1090
                        @122      EQU    $                                                     1091
02F1  E204 31B3                   LDR    D,4,0+DP                                              1092
02F3  80A4                        ADD    R,4,10                                                1093
02F4  E0B5                        LDR    R,5,11                                                1094
02F5  80A5                        ADD    R,5,10                                                1095
02F6  E356 2000                   LDR    DX,6,0+BUF,5                                          1096
02F8  9B46 2000                   STR    DX,6,0+BUF,4                                          1097
                        .                        BUF(.(@BP+@I).)=0                             1098
02FA  E0B6                        LDR    R,6,11                                                1099
02FB  80A6                        ADD    R,6,10                                                1100
02FC  9B6C 2000                   STR    DX,12,0+BUF,6                                         1101
                        .                        I=@I+1                                        1102
02FE  E0A6                        LDR    R,6,10                                                1103
02FF  6016                        ADD    IS,6,1                                                1104
0300  E06A                        LDR    R,10,6                                                1105
                        .                       END                                            1106
                        @123      EQU    $                                                     1107
0301  C903 02F1                   IBN     I,3,@122                                             1108
                        .                      DP=@DP+5                                        1109
0303  E2C6 31B3                   LDR    D,6,0+DP                                              1110
0305  6056                        ADD    IS,6,5                                                1111
0306  9C06 31B3                   STR    D,6,0+DP                                              1112
                        .                      END                                             1113
                        .          END                                                         1114
```

```
LOC   OBJECT CODE      CARD IMAGE                                                    CARDNUM
                      @121      EQU   $                                               1115
                      @118      EQU   $                                               1116
                      .    END                                                        1117
0308  C107 02DD                 BRC     I,7,@113                                      1118
                      .END MAIN                                                       1119
030A  BF17                      BRC     R,7,1                                         1120
                      .                                                               1121
                      .                                                               1122
                      ./"****   LINK MODULE FOR EXECUTING PL/ATAC PROGRAM.            1123
                      .         $EXECUTIVE                                            1124
                      .****"/                                                         1125
                      .    MODULE LINK                                                1126
                                ENTRY LINK                                            1127
                      .        DECLARE PROCSTK(64)RAM                                 1128
                                CRG   RAM                                             1129
3241                  PROCSTK   DS    65                                              1130
                      .       RO = PROCSTK                                            1131
                                ORG   ROM                                             1132
                      LINK      EQU   $                                               1133
0308  400C                      LDR   IS,12,0                                         1134
030C  E100 3241                 LDR   I,0,0+PROCSTK                                   1135
                      .       CALL MAIN()                                             1136
030E  601C                      ADD   IS,0,1                                          1137
030F  8DA0                      STK   DR,0,11                                         1138
0310  EDC1 00F9                 BAL   I,1,0+MAIN                                      1139
0312  8EA0                      RET   DR,0,11                                         1140
0313  6FFC                      ADD   IS,0,-1                                         1141
0314  5C02                      LDR   RX,2,0,12                                       1142
                      .    END LINK                                                   1143
0315  BF17                      BRC     R,7,1                                         1144
                      .EOF EOF EOF EOF EOF EOF EOF EOF EOF                            1145
                                END                                                   1146
```

```
     DUMP OF PSEUDO-MEMORY SEGMENT ROM  FROM 0000 TO 1FFF
0000     FFFF 0318 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
0010     400C 4022 9C02 3184 4012 9C02 3187 9C02 3186 9C0C 3185 E202 3185 2022 C106 0022
0020     C107 0101 E202 3184 6FF2 E324 3089 D8F3 E205 3186 FC53 E205 3184 E357 3089 08F6
0030     E208 3187 FC86 6FE7 E208 3184 E383 3089 D8F2 E205 3187 FC52 6023 E035 E056 8476
0040     8854 C106 0051 E205 3185 6015 9C05 3185 E205 3186 6015 9C05 3186 C107 0018 C107
0050     005F C103 005F E205 3185 6015 9C05 3185 E205 3187 6015 9C05 3187 C107 0018 E205
0060     3184 6015 9C05 3184 E205 3187 9C05 3186 4015 9C05 3187 E205 3184 2045 C101 0072
0070     C107 0018 4015 9C05 3184 9C0C 308F 4FB5 C107 0087 E206 3184 E207 308F 8367 30B9
0080     9C07 308F E207 3184 6017 9C07 3184 C905 007A E207 308F 9C07 3188 E207 3185 6047
0090     E203 308F D8F2 FC72 9C03 308F E202 3185 6042 E204 3188 D8F3 FC23 E204 3185 6044
00A0     E045 D8F4 4026 FC64 8063 C106 00AC E206 308F 6016 9C06 308F E0A6 E367 2FA0 9C07
00B0     3183 E207 30C1 6067 9C07 30C1 E0A8 D8F7 4052 FC27 E202 30C1 9882 3188 E202 30C1
00C0     6012 E203 3183 6013 E334 2000 9824 30C2 E204 30C1 6024 E0A6 D8F5 4057 FC75 CD06
00D0     0002 E367 3068 9847 30C2 E207 30C1 6037 E0A3 D8F2 4054 FC42 CD03 0002 6013 E334
00E0     3068 9874 30C2 E204 30C1 6044 E205 3183 6045 E356 2000 9846 30C2 E206 30C1 6056
00F0     E207 308F 9867 30C2 E0A8 D8F7 4052 FC27 4012 9882 3163 E202 3088 6012 9C02 3088
0100     BF17 E0A3 D8F2 4054 FC42 4014 9834 3138 BF17 400C 4FA2 9C02 30C1 E102 0460 9C02
0110     31E2 E0CA E0CB 4012 9C02 3088 E102 FF37 C107 0121 E0A3 4FF4 9834 2FA0 E0A4 6014
0120     E04A C902 011A E0CA E0A4 984C 2FA0 FE01 012A 0017 2000 2F9F E084 6024 E345 2000
0130     9C05 31E7 E085 6035 E356 2000 9C06 31E8 BC1E 31E7 A90E 31E2 9C1E 3238 E0A7 D8F6
0140     4C58 FC86 CD07 0002 9C07 31E6 E208 31E6 E202 3236 9882 31E9 E202 31E6 6012 E203
0150     323C 9823 31E9 E203 31E6 E204 31E7 9834 3068 E204 31E6 6014 E205 31E8 9845 3068
0160     ECB5 6055 E05B E0B5 8105 0F9F C106 016A C107 02DD E0B5 E356 2000 C105 0171 C107
0170     0160 E0CA E0C9 E0A6 B106 00C3 C106 017A C107 02C3 E0A6 E367 2FA0 C105 0184 E0A7
0180     6057 E07A C107 0173 E087 6027 E378 2000 9C08 323D E088 6038 E382 2000 9C02 323E
0190     E0A3 D8F2 4054 FC42 CD03 0002 E334 31E9 9C04 323F E0A5 D8F4 4056 FC64 CD05 0002
01A0     6015 E356 31E9 9C06 3240 8C1E 323D BC17 323F AAE7 9C17 3239 E206 3239 C101 021B
01B0     E0A7 D8F6 4058 FC86 E378 3163 C106 01FE E208 3088 6FF8 9C08 3088 E0A3 D8F2 4054
01C0     FC42 9B3C 3163 E0A4 E345 2FA0 E356 2000 9C06 31B9 9C0C 31E4 4FA6 C107 01E9 9C0C
01D0     31E5 4FA7 C107 01E2 E0A8 8208 31E4 E382 2FA0 8202 31E5 982C 2000 E202 31E5 6012
01E0     9C02 31E5 C907 01D4 E202 31E4 6012 9C02 31E4 C906 01CF E0A3 D8F2 4054 FC42 E334
01F0     318B E0A6 D8F5 4057 FC75 E207 31B9 8367 31BA D8F6 4028 FC86 9847 30C2 E0A3 D8F2
0200     4054 FC42 983C 3138 9C0C 31E4 4FA4 C107 0214 E0A5 8205 31E4 4FF6 9856 2FA0 E206
0210     31E4 6016 9C06 31E4 C904 0209 E0A6 6056 E06A C107 0173 E086 6016 E0A7 E378 2FA0
0220     6018 E362 2000 8382 2000 C102 022C E0A2 6052 E02A C107 0173 E082 6042 E0A3 E334
0230     2FA0 6044 E325 2000 8345 2000 C102 023D E0A5 6055 E05A C107 0173 E0A6 D8F5 4057
0240     FC75 E367 3163 C102 0261 E0A8 D8F7 4052 FC27 E0B2 E325 2000 9883 313A 9C0C 31E4
0250     4FA3 C107 025D E084 8204 31E4 984C 2000 E204 31E4 6014 9C04 31E4 C903 0253 C107
0260     0160 E0A5 D8F4 4056 FC64 E356 3138 2016 C105 026C C107 0160 4FC6 C107 027E E097
0270     6017 E079 E0A7 8097 E378 2FA0 C103 027E E0A8 8098 9888 2FA0 C107 0160 C906 026F
0280     E0A8 6048 9888 2FA0 E208 3088 20A8 C105 0292 E0A3 D8F2 4054 FC42 4014 9834 3138
0290     C107 0160 4019 4FB4 C107 0286 E095 E0A6 8096 E367 2FA0 6027 E0A8 8098 6FF8 E382
02A0     2FA0 6022 E373 2000 8723 2000 9853 3089 E093 E335 3089 C103 02B5 E095 E096 E367
02B0     3CB9 8107 7FFF 9857 3089 E097 6017 E079 C904 0296 6010 8DA0 ED01 0010 8EA0 6FF0
02C0     5C07 C107 0160 E0CA E0A7 E378 2FA0 C103 02CE E0A8 9888 2FA0 C107 012C E0A8 6058
02D0     E08A B108 00C3 C106 02DB FE00 02D8 0017 2FA0 3063 BF17 C107 02C4 E0CB E0B8 6018
02E0     E382 2000 C105 02E8 9C0B 31B3 C107 02ED E0B2 6052 E02B C107 02DE E0B2 6052 E02B
02F0     ECB2 B102 0F9B C106 02F6 BF17 E0B2 6012 E323 2000 2003 C102 0318 E0CA 4FA3 C107
0300     0311 E204 31B3 80A4 E085 80A5 E356 2000 9846 2000 E0B6 80A6 986C 2000 E0A6 6016
0310     E06A C903 0301 E206 31B3 6056 9C06 31B3 C107 02ED BF17 400C E100 3241 6010 8DA0
0320     ED01 0109 8EA0 6FF0 5C02 BF17 0000 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
0330     FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
       LINES 0340 THROUGH 1FE0 SAME AS ABOVE.
1FF0     FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
```

```
DUMP OF PSEUDO-MEMORY SEGMENT RAM  FROM 2000 TO 37FF
2000     0119 0010 1D16 0000 0016 0119 0040 1D1D 0000 000A 0119 0030 1D1E 0000 000A 0119
2010     001A 1D2E 0000 000F 0119 0010 1D3A 0000 0016 0119 0040 1D38 0000 0014 0119 0030
2020     1D4F 0000 0012 0119 0040 1080 0000 0014 0119 0030 1D8A 0000 000A 0119 0010 1DC6
2030     00C0 0016 011A 0040 1DDC 0000 000A 011A 001A 10EF 0000 000F 011A 0030 1DFE 0000
2040     0012 011A 0010 1E0A 0000 000E 011A 0014 1E11 0000 000F 011A 0040 1E26 0000 0014
2050     011A 0014 1E4C 0000 000F 011A 0010 1E53 0000 0016 011A 0030 1E56 0000 000A 011A
2060     0010 1E58 000C 000E 011A 0027 1E63 0000 0018 011A 0010 1E76 0000 0016 011A 0014
2070     1E66 0000 000F 011A 001A 1E97 0000 000F 011A 0010 1E99 0000 0016 011A 0040 1E9B
2080     0000 0014 011A 0040 1E9B 0000 000A 011A 001D 1EA6 0000 000E 0118 0030 1EAE 0000
2090     0012 011B 0014 1EC1 0000 000F 011B 0030 1EF3 0000 000A 011B 0010 1EF4 0000 000E
20A0     011B 0014 1EFC 0000 000F 011B 0040 1F10 0000 0014 011B 0010 1F26 0000 0016 0118
20B0     0027 1F28 0000 0018 011B 0014 1F36 0000 000F 0118 001A 1F3C 0000 000F 011B 001D
20C0     1F42 0000 000E 0118 0010 1F49 0000 0016 011B 0040 1F5B 0000 000A 011B 0030 1F5E
20D0     00C0 0012 011B C010 1F6C 0000 0016 0118 0014 1F71 0000 000F 0118 0040 1F85 0000
20E0     0014 011C 0030 1F8F 0000 000A 011C 0010 1F8F 0000 0016 011C 0010 1F90 0000 000E
20F0     011C 0014 1FA8 0000 000F 011C 0010 1FDE 0000 000E 011C 001A 1FE0 0000 000F 011C
2100     0014 1FE6 0000 000F 011C 0027 1FE0 0000 0018 011C 0040 1FFA 0000 0014 011C 0030
2110     200E 00C0 0012 C11C 0040 201A 0000 000A 011C 0010 201C 0000 0016 011C 0014 2021
2120     00C0 C00F 011C 0030 202B 0000 000A 011C 001D 202D 0000 000E 011C 0010 203F 0000
2130     C016 011C 0014 205B 0000 000F 0110 0010 2062 0000 0016 011D 0040 2070 0000 0014
2140     C11D 0010 207B 0000 000E 0110 0014 2096 0000 000F 0110 001A 2098 0000 000F 011D
2150     C027 2083 0000 0018 011D 0030 20BE 0000 0012 011D 0030 20C7 0000 000A 011D 001D
2160     20C9 0000 000E 011D 0014 2000 0000 000F 0110 0040 200A 0000 000A 0110 0040 20E5
2170     0000 0014 011D 0010 20EF 0000 0016 011D 0014 2108 0000 000F 011D 0010 2112 0000
2180     0016 011D 001D 2117 0000 000E 011D 0010 2135 0000 0016 011E 0014 2145 0000 000F
2190     011E 0030 2164 0000 000A 011E 0010 2165 0000 000E 011E 0027 2178 0000 0018 011E
21A0     0014 2180 0000 000F 011E 001D 2183 0000 000E 011E 0014 2188 0000 000F 011E 0014
21B0     21F5 0000 000F 011E 0063 2298 0000 0063 0000 0000 0000 0000 0000 0000 0000 0000
21C0     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
      LINES 21D0 THROUGH 2F90 SAME AS ABOVE.
2FA0     0708 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
2FB0     FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
      LINES 2FC0 THROUGH 2FE0 SAME AS ABOVE.
2FF0     FFFF FFFF FFFF FFFF FFFF 0190 02CB 0415 0555 0688 FFFF FFFF FFFF FFFF FFFF FFFF
3000     FFFF FFFF FFFF FFFF FF6F FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
      LINES 3010 THROUGH 3C50 SAME AS ABOVE.
3060     FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 2298 0000 1CF0 0000 1D16 0000 1010 0000
3070     1D1E 0000 1D2E C000 1D3B 0000 1D4F 0000 1D71 0000 107B 0000 1D86 0000 1D9B 0000
3080     1084 0000 1DEE 0000 1DFF 0000 1E0A 0000 1E11 0000 1E63 0000 0000 0000 0000 0000
3090     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
      LINES 30A0 THROUGH 3CA0 SAME AS ABOVE.
30B0     0000 0000 0000 0000 0000 0000 0000 0000 0002 0000 00C5 00C5 00C6 00C5 0000 00C5
30C0     C000 0036 011B C029 1CFD 0000 000A 0028 011B 0035 1D71 0000 0005 0013 0118 001D
30D0     107B 0000 0029 000C 0118 0045 1086 0000 0007 001E 011B 0017 109B 0000 0032 0031
30E0     011A 0020 1CD5 C000 0019 0069 011C 0034 1DFF 0000 0010 002F 011C 0028 10EE 0000
30F0     C00A 002B 0118 003D 1DB4 0000 0032 004E 0000 0027 1E63 0000 0018 00C5 0000 0000
3100     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
      LINES 3110 THROUGH 3160 SAME AS ABOVE.
3170     C000 0000 0000 C000 0001 0000 0C00 0000 0000 0000 0000 0000 0000 0000 0000 0000
3180     C000 0000 0000 0000 0000 0000 0300 0000 0000 0000 0000 001E 0000 0000 0000 0000
3190     0000 0000 0000 C006 000C 0012 0C18 0030 002A 0024 0000 0000 0036 0000 0000 0000
31A0     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
31B0     00C0 0000 0000 01B8 0005 0000 0001 0001 0315 011A 0110 011E 0000 0000 0000 0000
31C0     0000 0000 011E 011E 011E 011E 011E 011E 011E 0000 0000 0000 0000 0000 0000 0000
31D0     0000 0000 0000 3000 0C00 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

```
31E0     0000 0000 0460 0000 0005 0005 0000 2298 0000 26F8 0000 215D 0000 2176 0000 217D
31F0     0000 217E 0000 218E 0000 2198 0000 21AF 0000 21D1 0000 21DB 0000 21E6 0000 21FB
3200     0000 2214 000C 224E 0000 225F 0000 226A 0000 2271 0000 22C3 0000 0000 0000 0000
3210     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
       LINES 3220 THROUGH 3220 SAME AS ABOVE.
3230     0000 0000 0000 0000 0000 0000 0000 0000 0000 002B 0000 26F8 0000 2298 0000 22C3
3240     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0322 0557
3250     0004 0000 00C5 0059 0005 0058 0005 0055 068B 0000 0000 0000 0000 0000 0000 0000
3260     0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
       LINES 3270 THROUGH 3270 SAME AS ABOVE.
3280     0000 0000 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
3290     FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
       LINES 32A0 THROUGH 37E0 SAME AS ABOVE.
37F0     FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
```

BIBLIOGRAPHY

1. Frazer, P.D., The Electronic Warfare Application of Special Purpose Microprogrammed Minicomputers, M.S.E.E Thesis, Naval Postgraduate School, June, 1974.

2. ATAC, Applied Technology Airborne Computer, V. I,II, Itek Corporation, 1974.

3. Electronic Countermeasures, U.S. Army Signal Corps, 1961.

4. User's Manual, 1st ed., W.R. Church Computer Center, Naval Postgraduate School, Monterey, California, 1970.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 57 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	2
4.	Professor Stephen Jauregui, Jr., Code 52Ja Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	10
5.	Commander Navy Security Group Command (G-80) Navay Security Group Headquarters 3801 Nebraska Avenue, N.W. Washington, D.C. 20390 ATTN: CDR H. Orejuela	2
6.	Commander Naval Electronics Systems Command (PME 107) Naval Electronics Systems Command Headquarters Washington, D.C. 20360 ATTN: LCDR A. Sagarian	2
7.	Commander Naval Electronics Systems Command (PME 107) Naval Electronics Systems Command Headquarters Washington, D.C. 20360 ATTN: LCDR R. Shields	1
8.	Commander Naval Air Systems Command Naval Air Systems Command Headquarters Washington, D.C. 20361 ATTN: Air-53321A CDR V. McCauley	1

9. Applied Technology 1
645 Almanor Avenue
Sunnyvale, California 94086
ATTN: Mr. J. Houston

10. LT Robert D Stanga, Jr. 1
1105 Sparrow Drive
Pt. Mugu, California 93042